A. Pusztai Gh. Ardelean

LaTeX

Ghid de utilizare

EDITURA TEHNICĂ

LaTeX

Ghid de utilizare

$L^{A}T_{E}X$
User's Guide

If you want to get the fastest start using $L^{A}T_{E}X$ *editor that runs on IBM PCs and compatibles, this book should be at your side. We present you a step-by-step guide that includes exercices, practical examples and clear illustrations to help you gain confidence in no time. Beginning users will grasp the basics faster, while familiar users willl find new ways to take advantage on the program's special features.*

TEX este marcă înregistrată a AMS (American Mathematics Society)

Artur Pusztai **Gheorghe Ardelean**

LaTeX

Ghid de utilizare

Editura Tehnică
Bucureşti

Copyright © 1994, Editura Tehnică
Toate drepturile asupra acestei ediții
sunt rezervate editurii

Adresa: EDITURA TEHNICĂ
Piața Presei Libere, 1
33 Bucureşti, România
cod 79738

Redactor: ing. Daniela Raduly
Tehnoredactor: Victoria Ungureanu
Coperta: ing. Daniela Raduly

Coli de tipar: 11; Bun de tipar: 5.12.1994
C.Z. 519.682
ISBN 973-31-0712-3

Cuvânt înainte

De multe ori puterea nu vine atât din posesiune sau stăpânire ci din capacitatea de a face ceva ce alţii nu ştiu sau nu pot. Astăzi ni se oferă posibilităţi nebănuite de a deveni mai puternici prin ceea ce facem cu ajutorul unui calculator bun şi a unor programe inteligent concepute.

Această carte vă va familiariza cu cel mai performant program de formatat texte. Cu ajutorul lui veţi putea face ca din imprimanta calculatorului dumneavoastră să apară texte editate conform celor mai exigente standarde de tehnoredactare.

Dacă nu veţi avea pretenţia unor tiraje mari, vă veţi simţi ca şi cum aţi stăpâni o editură şi aţi poseda o tipografie.

dr. I. Cotăescu

Prefaţă

TEX este un sistem de culegere de texte
destinat creării unor cărţi frumoase —
şi în special a celor care conţin multă matematică.

– D. E. KNUTH, The TEXbook (1983)

LATEX este o extensie a lui TEX, care îl face mai uşor de utilizat pentru scrierea unor documente standard: articole, rapoarte, cărţi, scrisori. Această carte este adresată atât celor care nu au lucrat niciodată cu LATEX cât şi celor care au deja o experienţă în utilizarea acestui program.

Cu riscul de a nemulţumi cititorii din ambele categorii, am optat pentru o creştere treptată a dificultăţii capitolelor, în aşa fel încât cititorii care învaţă LATEX să poată folosi această lucrare ca pe un manual.

Totuşi, nu am dorit să revenim de prea multe ori la o anumită temă. De aceea, unele paragrafe sunt tipărite cu litere mai mici şi sunt precedate de semnul:

Acestea de regulă folosesc noţiuni care vor fi definite mai târziu, dar locul lor este totuşi acolo unde apar. Dacă sunteţi la prima lectură a cărţii, treceţi peste aceste pasaje — veţi putea reveni mai târziu.

•

Am încercat să construim un index cât mai bogat, în aşa fel încât să fie într-adevăr folositor. Dacă vă interesează o anumită comandă sau cuvânt

cheie LaTeX, căutaţi în index. Dacă însă căutaţi detalii despre o anumită temă, consultaţi mai întâi cuprinsul şi abia dacă nu aţi găsit aici nici un titlu apropiat de ceea ce vă interesează, apelaţi la index.

Multe comenzi simple nu sunt cuprinse în index. În lista tabelelor veţi găsi însă cu siguranţă numele unui tabel care să conţină comanda căutată.

Dacă un număr de pagină din dreptul unei intrări în index este scris cu caractere italice, înseamnă că la pagina respectivă veţi găsi un exemplu pentru utilizarea acelui obiect.

Autorii.

Cuprins

1 Introducere **1**

1.1 Ce este LaTeX? 1
1.2 Un exemplu 2
1.3 De ce LaTeX? 4
1.4 Cum se foloseşte LaTeX? 5
1.4.1 Editarea fişierului sursă 5
1.4.2 Procesarea fişierului sursă 6
1.4.3 Vizualizarea documentului 7
1.4.4 Tipărirea documentului 7

2 Noţiuni de bază **9**

2.1 Secvenţe de control 10
2.2 Modul matematic 12
2.3 Adaptarea LaTeX-ului la alte limbi 13
2.3.1 Accente 13
2.3.2 Titluri 14
2.3.3 Despărţirea în silabe 15

3 Organizarea logică a documentului **17**

3.1 Structura generală a fişierului de intrare 17
3.1.1 Comanda `\documentstyle` 17
3.1.2 Comanda `\begin{document}` 19
3.1.3 Comanda `\end{document}` 19
3.2 Construirea titlului documentului 19
3.2.1 Comanda `\title` 19
3.2.2 Comanda `\author` 20
3.2.3 Comanda `\date` 20

3.2.4 Comanda \maketitle 20
3.2.5 Trimiteri la note de subsol 21
3.3 Capitole şi secţiuni 21
3.3.1 Comanda \abstract 22
3.3.2 Comanda \part 23
3.3.3 Comanda \chapter 23
3.3.4 Comenzile \section, \subsection şi \subsubsection 23
3.3.5 Comanda \appendix 24
3.4 Etichete şi referiri 24
3.4.1 Etichetarea 25
3.4.2 Referirea 25
3.5 Cuprinsul 25

4 Cadruri **27**
4.1 Documentul 27
4.2 Rezumatul (Abstract) 28
4.3 Citate 28
4.3.1 Cadrul quote 28
4.3.2 Cadrul quotation 29
4.3.3 Cadrul verse 29
4.4 Alinieri 29
4.5 Păstrarea nemodificată a textului 30
4.6 Liste 31
4.6.1 Cadrul itemize 31
4.6.2 Cadrul enumerate 32
4.6.3 Cadrul description 32
4.6.4 Cadrul list 33
4.7 Tabele 34
4.7.1 Cadrul tabbing 34
4.7.2 Cadrul tabular 35
4.8 Ecuaţii 37
4.8.1 Cadrul displaymath 37
4.8.2 Cadrul equation 38
4.8.3 Cadrul array 38
4.8.4 Cadrul eqnarray 39
4.9 Desene 40
4.10 Cadruri mobile 42
4.10.1 Cadrul table 43
4.10.2 Cadrul figure 44

4.10.3 Pagini miniaturale 46
4.11 Bibliografia 46
4.12 Index şi glosar 48

5 Macrodefiniţii 51

6 Modul paragraf 61
6.1 Tipuri şi dimensiuni de caractere 61
6.2 Simboluri speciale 63
6.3 Gestionarea rândurilor şi a paginilor 66
6.3.1 Gestionarea rândurilor 66
6.3.2 Gestionarea paginilor 67
6.4 Spaţii şi spaţieri 68
6.5 Note marginale 70
6.6 Dreptunghiuri pline 71

7 Modul matematic 73
7.1 Comenzi în mod matematic 74
7.2 Spaţii în mod matematic 80
7.3 Caractere în mod matematic 82

8 Modul LR 85

9 Variabile interne LaTeX 89
9.1 Lungimi 89
9.1.1 Lungimi pentru controlul *layout*-ului 90
9.1.2 Designul unei pagini 91
9.2 Variabile întregi 95
9.3 Parametrii de stil 97
9.3.1 Ecuaţii matematice 97
9.3.2 Cadrurile `array` şi `tabular` 98
9.3.3 Cadrurile `table` şi `figure` 98

10 Erori şi avertismente 101
10.1 Mesaje de eroare 102
10.2 Mesaje de avertisment 104

ANEXE

A Lucrul cu fişiere incluse **107**

B Definirea opţiunii `romanian` **109**

C Caractere ASCII **113**

D Stilul *letter* **115**

E SLiTEX **119**

F REVTEX **123**

G Distribuţia emTEX **131**

G.1 `dvidrv` 132
G.2 Modul de rulare al utilitarelor 134
G.3 Opţiuni comune tuturor acestor utilitare 134
G.3.1 Opţiuni pentru afişarea de mesaje 135
G.3.2 Opţiuni de selectare a paginilor 136
G.3.3 Opţiuni referitoare la font 136
G.3.4 Opţiuni referitoare la *layout*-ul paginii 137
G.3.5 Colajul 139
G.3.6 Opţiuni referitoare la rezoluţie şi mărire 140
G.3.7 Opţiuni referitoare la directoare şi nume de fişier . . . 141
G.3.8 Alte opţiuni 142
G.4 Opţiuni suplimentare pentru `dviscr` şi `dvivik` 143
G.5 Opţiuni suplimentare pentru `dvimsp` 145
G.6 Opţiuni suplimentare pentru `dvidot` 145
G.7 Opţiuni suplimentare pentru `dvihplj` 146
G.8 Taste funcţionale pentru `dviscr` şi `dvivik` 146
G.9 Linia de stare pentru `dviscr` şi `dvivik` 149

Index **151**

Bibliografie **153**

Lista tabelelor

4.1 **Un mic exemplu** 45

6.1 **Clase de fonturi** 63
6.2 **Dimensiunea fonturilor.** 64
6.3 **Accente în mod paragraf** 65
6.4 **Simboluri speciale (orice mod)** 65
6.5 **Simboluri speciale (numai în mod paragraf)** 65

7.1 **Funcţiile matematice** 75
7.2 **Delimitatori** 75
7.3 **Accente în mod matematic** 78
7.4 **Alte simboluri în mod matematic** 78
7.5 **Operatori mari** 78
7.6 **Operatori binari** 79
7.7 **Operatori relaţionali** 79
7.8 **Operatori de negaţie** 79
7.9 **Simboluri de tip săgeată** 80
7.10 **Comenzi sinonime** 80
7.11 **Literele mici ale alfabetului grecesc** 81
7.12 **Literele mari ale alfabetului grecesc** 81

Capitolul 1

Introducere

1.1 Ce este LaTeX?

Pentru a publica o lucrare, autorul predă editorului un manuscris. Editorul va trebui să verifice dacă conţinutul semantic al lucrării corespunde cerinţelor sale, şi rezolvă acest lucru prin alegerea unuia sau a mai multor recenzori, care vor citi manuscrisul şi vor verifica corectitudinea conţinutului.

Pe baza referatelor recenzorilor, editorul va accepta lucrarea, o va respinge, sau îi va propune autorului să facă schimbările sugerate de recenzori, urmând ca manuscrisul corectat să fie din nou recenzat.

Odată acceptată spre a fi publicată, lucrarea trebuie aranjată în pagină. Acest lucru îl face tehnoredactorul, care hotăreşte aspectul şi dimensiunile paginii, fonturile etc.. După ce acesta a stabilit *layout*-ul lucrării de publicat, zeţarul culege textul urmând acest model, şi va scoate o primă copie care va fi supusă corecturilor sintactice (gramaticale). Pe baza acestor corecturi, zeţarul face şi el modificările, după care se tipăreşte lucrarea în numărul de exemplare hotărât de editor.

Recenzia este făcută întotdeauna de oameni. Toate celelalte etape pot fi efectuate cu ajutorul calculatorului, ele având reguli fixe şi relativ simple.

LaTeX este tehnoredactorul, iar TeX este zeţarul în scenariul prezentat mai sus.

TeX (se pronunţă *tehi*, se poate scrie şi TeX) este un program elaborat de Donald E. Knuth [1], care foloseşte la culegerea şi tipărirea de texte şi formule matematice.

LaTeX (se pronunţă *la-tehi* sau *lei-tehi*, se poate scrie şi LaTeX) este construit de Leslie Lamport [2] pe baza lui TeX, definind aşa-numite *layout*-uri

(LAyout for TEX) pentru cărţi, articole, rapoarte şi scrisori.

LATEX este un pachet de macrodefiniţii pentru TEX, iar TEX este un sistem de prelucrare a textelor. LATEX şi TEX nu sunt editoare de texte!

1.2 Un exemplu

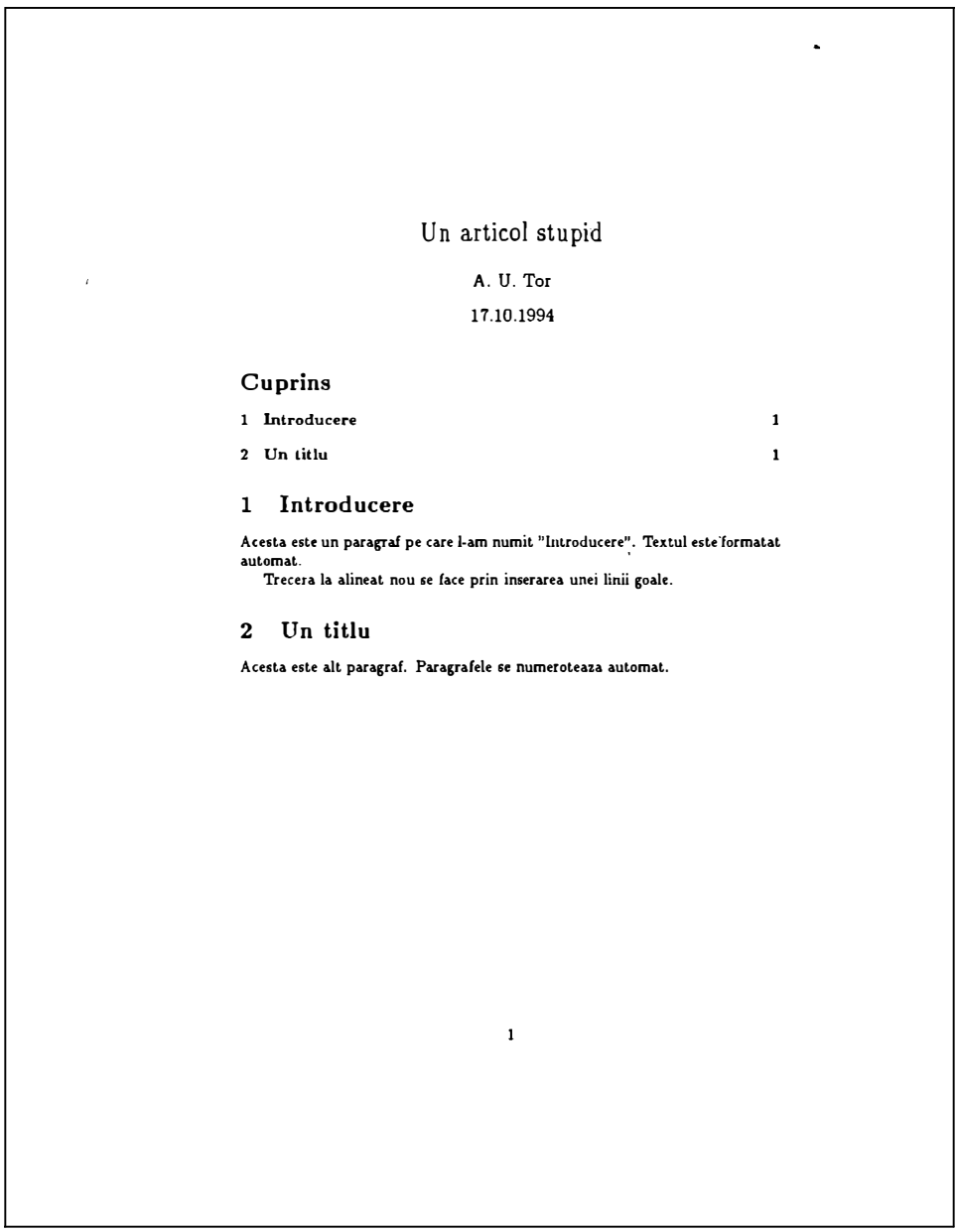
Când tehnoredactorul lucrează la *layout*, el recunoaşte în text intenţiile autorului: ce este titlu de capitol, ce este titlu de paragraf, ce este citat, ce este listă etc.. LATEX nu poate recunoaşte aceste lucruri fără ca autorul să-i dea informaţii suplimentare în legătură cu textul ce-l scrie. În figura **1.1** este dat un exemplu de fişier LATEX minimal, dat aici numai pentru exemplificare. În figura **1.2** este dat rezultatul acestui fişier, în urma procesării lui.

Cuvintele care încep cu caracterul \ sunt comenzi LATEX, care nu vor apărea în documentul final. Toate caracterele de la un caracter % şi până la sfârşitul liniei vor fi ignorate de LATEX (sunt aşa-numitele comentarii).

```
\documentstyle{article} % dorim sa scriem un articol.
% implicit titlul cuprinsului este "Contents".
% Putem schimba acest lucru:
\renewcommand{\contentsname}{Cuprins}
% setam titlul, autorul si data:
\title{Un articol stupid}
\author{A. U. Tor}
\date{17.10.1994}
\begin{document}        % inceputul documentului.
\maketitle              % tipareste titlul, autorul si data.
\tableofcontents        % tipareste cuprinsul lucrarii.
\section{Introducere}   % inceputul unei sectiuni (paragraf).
Acesta este un paragraf pe care l-am numit "Introducere".
Textul este formatat automat.

Trecera la alineat nou se face prin inserarea unei linii goale.
\section{Un titlu}      % inceputul unui alt paragraf.
Acesta este alt paragraf. Paragrafele se numeroteaza automat.
\end{document}          % sfarsitul articolului.
```

Figura 1.1: Un exemplu simplu de articol scris în LATEX. Rezultatul procesării acestui fişier este dat în figura 1.2.



Un articol stupid

A. U. Tor

17.10.1994

Cuprins

1 Introducere 1

2 Un titlu 1

1 Introducere

Acesta este un paragraf pe care l-am numit "Introducere". Textul este formatat automat.

Trecera la alineat nou se face prin inserarea unei linii goale.

2 Un titlu

Acesta este alt paragraf. Paragrafele se numeroteaza automat.

1

Figura 1.2: Rezultatul procesării fişierului sursă din figura 1.1. Chenarul negru reprezintă o coală A4.

1.3 De ce LaTeX?

Există multe editoare de texte, unele dintre ele chiar foarte performante, care desenează – să zicem – la fel de frumos o integrală sau un radical, sau care oferă la fel de multe fonturi ca LaTeX. Şi mai cu seamă, majoritatea lor sunt din categoria sistemelor *What You See Is What You Get*, ceea ce le face deosebit de apreciate.

LaTeX nu este un sistem *What You See Is What You Get*, ceea ce înseamnă că în timpul scrierii textului, autorul nu vede pe monitor efectul celor scrise. De fapt, autorul va folosi un editor oarecare pentru scrierea textului; el poate însă să întrerupă în orice moment editarea, să proceseze textul şi să vizualizeze fişierul rezultat (amănunte despre folosirea pachetului de utilitare asociate cu LaTeX sunt date în paragraful 1.4).

Atunci de ce LaTeX?

Răspunsul este: deoarece LaTeX ne obligă să ne concentrăm asupra organizării logice a lucrării, lăsând pe seama lui formatarea textului.

Tehnoredactarea unei cărţi (sau a altei lucrări) este o meserie care se învaţă în timp, şi nu oricine poate face acest lucru. S-ar putea crede că totul este ca aspectul lucrării să fie frumos, dar lucrurile nu stau chiar aşa. Cărţile sunt tipărite pentru a fi citite, şi deci mai important decât aspectul general este lizibilitatea şi claritatea textului. Organizarea grafică a textului trebuie să sugereze structura lui logică.

De asemenea, important este nu să se ocupe pagina cât mai uniform, ci ca citirea să se facă cât mai comod.

Anumite părţi ale textului vor trebui să iasă în evidenţă (de exemplu citatele). Dar felul în care acestea sunt evidenţiate nu este indiferent, şi mulţi ar alege modalitatea necorespunzătoare. În LaTeX trebuie specificat doar *ce* este partea respectivă a textului (titlu de capitol, citat, cuvânt important ce trebuie evidenţiat în cadrul textului etc.) iar LaTeX va alege forma potrivită pentru tipărire.

Iată cum se inserează scurte citate în text (ca regulă generală în toată lucrarea de faţă, când se dau exemple pe două coloane, în dreapta va fi textul sursă, iar în stânga efectul său):

O regula de aur a tipografilor spune:

> Nici un rand sa nu contina mai mult de 66 de caractere.

De aceea in ziare se folosesc mai multe coloane alaturate ...

```
O regula de aur a tipografilor spune:
\begin{quote}
Nici un rand sa nu contina mai mult de 66
de litere.
\end{quote}
De aceea, in ziare se folosesc mai multe
coloane alaturate \dots
```

Este important de asemenea să se folosească cât mai puţine fonturi posibil, deoarece cu cât sunt folosite mai multe fonturi, cu atât mai greoi de citit devine textul.

Acestea sunt doar câteva din problemele de care trebuie ţinut seama la proiectarea unui *layout*, dar chiar şi pe acestea am fi tentaţi să le ignorăm dacă ar trebui să ne ocupăm de design.

Folosind LaTeX însă, este foarte greu să modificăm aspectul general al documentului, rezultatul însă va fi profesional, deoarece în locul nostru L. Lamport a colaborat cu tehnoredactori de prima mână din Statele Unite pentru a ne oferi layout-uri pentru cărţi, articole, reviste şi scrisori. Singurul dezavantaj este faptul că se respectă convenţiile valabile în S.U.A., multe dintre ele nefiind valabile în Europa. Majoritatea lor pot fi însă rezolvate foarte uşor (de exemplu, vezi `\frenchspacing`).

1.4 Cum se foloseşte LaTeX?

TeX şi LaTeX sunt produse *shareware*, ceea ce înseamnă că au preţuri deosebit de scăzute, sau sunt gratuite dacă nu sunt folosite în scopuri comerciale. Amănuntele necesare sunt livrate odată cu distribuţia (pachetul de programe).

Ambele programe sunt foarte răspândite, şi există practic sub toate sistemele de operare (UNIX, DOS, OS/2 etc.), ceea ce constitue un mare avantaj datorită portabilităţii fişierelor sursă ale documentelor.

Într-o distribuţie de LaTeX sunt livrate de obicei foarte multe fişiere (fonturi, utilitare, documentaţii, exemple). Toate descrierile acestora se vor găsi în documentaţiile aferente. De asemenea, fiecare distribuţie este însoţită de o bogată documentaţie ce foloseşte la instalarea produsului.

În continuare vom prezenta paşii ce trebuie parcurşi pentru a obţine un document pe hârtie.

1.4.1 Editarea fişierului sursă

Cu un editor oarecare de texte (de exemplu: sub UNIX, `vi`; sub DOS, Norton Editor (`ne`), sau Wordstar în mod `nondocument` etc.) se construieşte un fişier cu sufixul `.tex`. Nu este obligatoriu acest sufix, dar prezenţa lui face lucrul în continuare mai comod. În cele ce urmează, să presupunem că am editat fişierul prezentat în figura 1.1 (pagina 2), şi l-am numit `exemplu.tex`.

1.4.2 Procesarea fişierului sursă

Dacă LaTeX este instalat corespunzător (urmând indicaţiile venite împreună cu distribuţia de care dispuneţi), atunci procesarea se va face pur şi simplu lansând LaTeX cu numele fişierului de procesat ca parametru. De exemplu, pentru procesarea fişierului `exemplu.tex` se va da comanda:

```
prompter> latex exemplu <ENTER>
```

Pe ecran se va tipări următorul text:

```
This is ...
**&lplain exemplu
(exemplu.tex
ILaTeX Version ...
(...\texinput\article.sty
Document Style 'article...
(...\texinput\art10.sty))
No file exemplu.aux.
No file exemplu.toc.
[1] [2] (art.aux) )
Output written on exemplu.dvi (2 pages, 3540 bytes).
Transcript written on exemplu.log.
```

unde ... semnifică informaţii dependente de versiune şi de instalare.

Efectul primei procesări a acestui fişier LaTeX este scrierea mai multor fişiere:

`exemplu.aux` este un fişier cu informaţii auxiliare despre entităţile logice din fişierul sursă, precum şi despre *etichete* (vezi `\label` şi `\bibitem`). Este folosit de LaTeX la a doua procesare pentru rezolvarea referinţelor încrucişate (vezi `\cite`, `\ref` etc.).

`exemplu.toc` conţine toate informaţiile necesare alcătuirii cuprinsului lucrării. Este generat numai dacă în fişierul `.tex` apare comanda `\tableofcontents` (Table Of Contents). **Atenţie:** *Cuprinsul va apărea efectiv abia la a doua procesare a fişierului.*

`exemplu.log` este fişierul ce conţine raportul despre procesarea fişierului `.tex`, eventual erorile întâlnite, precum şi informaţii statistice despre folosirea resurselor calculatorului.

`exemplu.dvi` este fişierul de ieşire, care conţine practic rezultatul procesării, într-un cod independent de perifericul pe care va fi tipărit (DeVice Independent).

Pentru documente mai complexe, este posibil să fie produse şi alte fişiere, cum ar fi:

`*.lof` este asemănător cu `*.toc`, fiind creat de comanda `\listoffigures` (List Of Figures). Când se cere lista figurilor, este obligatorie procesarea de două ori a fişierului `.tex`.

`*.lot` conţine informaţiile necesare generării listei de tabele. Este generat de comanda `\listoftables`. Şi în acest caz este nevoie de reprocesarea fişierului `tex` pentru obţinerea în fişierul `dvi` a listei de tabele.

`*.idx` este produs de comanda `\makeindex` în combinaţie cu comenzile `\index` şi conţine comenzile `\indexentry` aferente. Paşii necesari obţinerii efective a indexului în documentul final sunt prezentaţi în paragraful 4.12 (pag. 48).

`*.glo` este produs de comanda `\makeglossary`. Amănunte se pot găsi în paragraful 4.12 (pag. 48).

În cazul în care în fişierul `.tex` s-au strecurat greşeli, LaTeX le va raporta. Pentru corectarea lor, vezi capitolul 10.

1.4.3 Vizualizarea documentului

Există posibilitatea de a vedea pe ecran imaginea ce ar ieşi la imprimantă, sub DOS cu ajutorul comenzii `v` *numefisier*, iar sub X WINDOWS cu ajutorul comenzii `xdvi` *numefisier*. Comanda `v` foloseşte utilitarul `dviscr`, care este capabil să transpună informaţia din fişierul `dvi` pe ecran. Vizualizarea unui document este foarte utilă mai ales în faza de elaborare, deoarece autorul poate vedea pe ecran efectul celor scrise, de obicei găsind erori ce trebuiesc corectate înainte de a tipări documentul pe hârtie.

1.4.4 Tipărirea documentului

Pentru a tipări fişierul la imprimantă este nevoie de un utilitar care să poată comanda imprimanta. Sub sistemul de operare DOS, pentru tipărire la o imprimantă cu ace, utilitarul se numeşte `dvidot`, dar el are nevoie de mai mulţi parametri. De aceea, în general pentru fiecare imprimantă există o comandă distinctă. De exemplu, pentru o imprimantă EPSON FX 80 se

dă comanda `prtfx exemplu`. Pentru imprimante laser, utilitarul se numeşte `dvihplj`, iar comanda de tipărire pentru o imprimantă din seria HEWLETT PACKARD LJ este `prthplj exemplu`.

Sub sistemul de operare UNIX, situaţia este asemănătoare. Numele utilitarului care poate tipări fişiere `dvi` la imprimantă se va găsi în ghidul local al sistemului. Pentru sisteme ce dispun de o imprimantă *PostScript* există utilitarul `dvi2ps` care transformă fişierul `dvi` într-unul `ps`, care apoi poate fi trimis la o imprimantă *PostScript*.

Se observă că dacă apar referinţe încrucişate, sau se doreşte scrierea automată a cuprinsului, a listei de figuri sau tabele, sau a unui index, fişierul trebuie procesat de două sau chiar de trei ori, deoarece pentru aceste operaţii LaTeX are nevoie de fişiere care la prima procesare încă nu există şi deci nu poate rezolva cererile.

Capitolul 2

Noţiuni de bază

Aşa cum am văzut în capitolul anterior, LaTeX nu este un sistem "What you see is what you get". Mai degrabă putem privi LaTeX ca pe un limbaj de programare, în care se folosesc anumite "instrucţiuni" pentu a obţine anumite efecte.

Pentru LaTeX, următoarele caractere au o semnificaţie aparte, ele neputând fi folosite "ad literam":

```
$ & % # _ { } ~ ^ " \ | < >
```

Nu putem folosi aceste caractere în textul sursă al documentului pentru a le obţine în document. Cunoaştem deja semnificaţia caracterului %: ceea ce urmează după el până la capătul rândului se ignoră (este folosit pentru comentarii în textul sursă). Semnificaţiile celorlalte caractere speciale vor fi prezentate pe parcurs.

Câteva din aceste caractere pot fi obţinute ca atare cu ajutorul secvenţelor de control:

```
$ & % # _ { }                    \$ \& \% \# \_ \{ \}
```

De asemenea, nici caracterul "spaţiu" (*blank*) nu este interpretat întotdeauna ca atare, şi anume oricâte spaţii succesive sunt interpretate ca un singur spaţiu.

Caracterul "linie nouă" (*newline*) este echivalent cu un spaţiu dacă nu este urmat de alt caracter "linie nouă". Mai mult de două caractere "linie nouă" succesive sunt interpretate ca două astfel de caractere şi semnifică terminarea unui paragraf.

Trecerea la un alt paragraf se poate face şi cu secvenţa de control `\par`.

2.1 Secvenţe de control

Secvenţele de control sunt comenzi pentru LaTeX. LaTeX dispune de un set extrem de mare de comenzi foarte puternice. Folosirea acestor comenzi este de fapt obiectul acestei cărţi.

O secvenţă de control începe întotdeauna cu caracterul \ (numit "backslash"). De exemplu, secvenţa de control `\today` are ca efect scrierea datei de pe sistemul de calcul pe care se face procesarea, iar secvenţa de control `\%` produce caracterul %.

O secvenţă de control este compusă fie dintr-un *backslash* urmat de o succesiune de oricâte litere, fie dintr-un caracter *backslash* urmat de un singur caracter diferit de o literă.

Pentru secvenţele de control din prima categorie trebuie menţionat că LaTeX ignoră spaţiile de după o astfel de comandă. De exemplu, pentru scrierea "logo"-ului LaTeX există o comadă: `\LaTeX`:

LaTeXeste un program de tehnoredactare.

```
\LaTeX este un program de tehnoredactare.
```

Oricâte spaţii am adăuga după secvenţa de control `\LaTeX`, rezultatul ar fi acelaşi: lipirea 'logo'-ului de următorul cuvânt. Soluţia este oferită de o altă secvenţă de control, `\␣` (un *backslash* urmat de un spaţiu), care forţează un spaţiu:

LaTeX este un program de tehnoredactare.

```
\LaTeX\ este un program de tehnoredactare.
```

Am văzut că pentu LaTeX un singur caracter 'rând nou' este echivalent cu un spaţiu. De aceea, în textul sursă paragrafele trebuie separate printr-un rând gol. Ca alternativă, există comanda `\par` care realizează trecerea la paragraf nou:

Un paragraf ... Acelasi paragraf ...
Paragraf nou ...
Alt paragraf.

```
Un paragraf ...
Acelasi paragraf ...

Paragraf nou ...\par
Alt paragraf.
```

Ca orice alt sistem de prelucrare a textelor, LaTeX dispune de un set bogat de fonturi. Implicit, pentru scrierea textului se foloseşte un font numit roman. Selectarea unui alt font se face cu ajutorul unei secvenţe de control, de exemplu `\tt` pentru fontul numit `typewriter` (asemănător cu maşina de scris), `\bf` pentru fontul **bold face** etc. Vom reveni mai târziu asupra acestor comenzi.

La un moment dat există un singur font activ. Comenzile de mai sus schimbă fontul activ, astfel încât dacă dorim tipărirea doar a câtorva cuvinte într-un font şi revenirea la cel precedent, LaTeX ne oferă o modalitate în care nu trebuie să cunoaştem fontul curent:

... nu cunosc fontul curent, **dar pot scrie temporar** cu alt font.

```
... nu cunosc fontul curent, {\bf
dar pot scrie temporar} cu alt font.
```

Perechea de acolade {} formează un grup. Modificarea fontului se face local în acest grup, de aceea la părăsirea grupului se reface automat fontul curent de la intrarea în grup.

Noţiunea de grup oferă mult mai multe facilităţi decât schimbarea temporară a fontului. Cu mici excepţii, toate variabilele LaTeX sunt salvate pe stivă la intrarea într-un grup şi refăcute la ieşirea din grup.

Acoladele mai au o utilizare în afară de delimitarea grupurilor, şi anume la transmiterea parametrilor secvenţelor de control cu parametri. Prin parametru se înţelege orice şir de caractere între acolade sau un singur caracter în cazul în care acoladele lipsesc.

De exemplu, comada `\hspace` are nevoie de un parametru care specifică spaţiul orizontal ce trebuie inserat în poziţia în care apare comanda. Parametrul trebuie să fie un număr urmat de o unitate de măsură cunoscută de LaTeX (vom reveni mai târziu cu o listă completă a acestora).

Aici este un centimetru.

```
Aici\hspace{1cm}este un centimetru.
```

Acoladele ce delimitează un parametru nu formează un grup. De aceea, modificarea unei variabile LaTeX în cadrul unui parametru al unei comenzi se va menţine şi după executarea comenzii respective. Dacă dorim refacerea valorii după executarea comenzii, însăşi comada trebuie inclusă într-un grup!

2.2 Modul matematic

LaTeX oferă multe facilităţi pentru culegerea formulelor matematice. Acestea sunt culese într-un mod diferit de textele obişnuite. Când LaTeX culege o formulă matematică spunem că este în mod matematic, iar când culege text obişnuit, spunem că se află în mod text. Există comenzi care funcţionează doar în mod matematic, şi altele care funcţionează doar în mod text.

LaTeX este implicit în mod text. Intrarea în modul matematic se poate face în multe feluri. Cel mai simplu de scris sunt formulele care apar în cadrul paragrafului. Ele trebuie puse între două caractere $:

Deoarece $x = y$ putem spune ... `Deoarece $x=y$ putem spune ...`

În mod matematic, caracterul _ poate fi folosit pentu a pune indici inferiori, iar caracterul ^ pentru indici superiori. Aceste două caractere se comportă asemănător cu secvenţe de control cu un parametru, iar acest parametru este cules ca indice. Dacă indicele este un singur caracter, el se pune imediat după caracterul special _ sau ^, iar dacă indicele este format din mai mult de un caracter el trebuie pus între acolade:

Fie $x_1, x_2, ..., x_{n+1}$ un ... `Fie $x_1, x_2, ..., x_{n+1}$ un ...`

Adesea o formulă mai importantă trebuie pusă pe un rând separat, şi scrisă ceva mai mare. În acest caz se va folosi perechea $$ ... $$:

Fie

$$x_1, x_2, ..., x_{n+1}$$

un ...

```
Fie
$$x_1, x_2, ..., x_{n+1}$$
un ...
```

Modul matematic se comportă ca un grup, în sensul că la ieşirea din modul matematic orice variabilă LaTeX modificată în acest mod va primi valoarea pe care o avea înainte de intrarea în mod matematic.

În textele matematice se folosesc adesea literele alfabetului grecesc. Acestea se pot obţine în mod matematic prin secvenţe de control de genul `\alpha`, `\beta` etc..

Pentru scrierea unei fracţii există secvenţa de control `\frac` care are doi parametri: primul parametru este numărătorul, iar al doilea numitorul:

... avem egalitatea:

$$\frac 14=\frac{1}{4}=\frac{\frac12}\alpha$$

```
... avem egalitatea:
$$\frac 14=\frac{1}{4}=\frac{\frac12}
\alpha$$
```

Am ilustrat aici faptul că în absenţa acoladelor care să delimiteze un parametru, este luat ca parametru primul caracter sau prima secvenţă de control.

De asemenea foarte des utilizaţi în scrierea formulelor matematice sunt operatorii, cum ar fi `\sum` pentru $\sum$, `\prod` pentru $\prod$, `\int` pentru $\int$ etc.. O listă completă a acestor operatori va fi dată mai târziu. Pentru scrierea limitelor se vor folosi indici. Indicii operatorilor vor fi poziţionaţi diferit la formule în text faţă de formule dispuse pe un rând separat:

... $\sum_{i=1}^n x_i = 0$ ...

$$\sum_{i=1}^n x_i = 0$$

```
... $\sum_{i=1}^nx_i=0$ ...
$$\sum_{i=1}^nx_i=0$$
...
```

La unii operatori, cum ar fi `\int` pentru $\int$, nu există această diferenţă, indicii rămânând indici. Pentru scrierea lor ca limite se va plasa comanda `\limits` imediat după respectivul operator. Şi viceversa, se poate folosi comanda `\nolimits` pentru a inhiba poziţionarea indicilor ca limite ale operatorului.

Deosebit de utile sunt şi secvenţele de control `\left` şi `\right` care se pot aplica pe paranteze pentru ajustarea automată a dimensiunilor acestora. Aceste două comenzi trebuie să fie perechi.

$$x+2\left(\frac{\sin x}{x}+\cos\left(x+1\right)\right)=0$$

```
$$x+2\left(\frac{\sin x}{x}+\cos\left(x+1
\right)\right)=0$$
```

2.3 Adaptarea LaTeX-ului la alte limbi

2.3.1 Accente

În limba română se folosesc în plus faţă de literele alfabetului latin literele: Â, â, Ă, ă, Î, î, Ş, ş, Ţ şi ţ. Pentru obţinerea lor, se folosesc *accente*. LaTeX oferă un set de secvenţe de control pentru obţinerea accentelor, în aşa fel încât poate fi folosit practic pentru orice limbă care are la bază alfabetul latin. Aceste secvenţe de control acceptă un parametru, care este de regulă

litera pe care se pune accentul. Lista completă a accentelor va fi dată mai târziu, pentru moment prezentăm doar cele de care avem nevoie pentru limba română:

Â, â, Ă, ă, Î, î, Ş, ş, Ţ, ţ.
Atenţie: NU Î. Accentul se aplică semnului ı(i fără punct).

```
\^A, \^a, \u A, \u a, \^I, \^\i, \c S,
\c s, \c T, \c t.\par
Aten\c tie: NU \^i. Accentul se aplic\u a
semnului \i (i f\u ar\u a punct).
```

Scrierea aceasta a textelor în limba română presupune un oarecare efort. De aceea, ar fi poate mai bine ca în acest moment să studiaţi anexa B în care este prezentat un fişier de definiţii menite să uşureze scrierea textelor în limba română. Totuşi, în toate exemplele din această carte se vor folosi comenzile prezentate mai sus pentru scrierea accentelor.

2.3.2 Titluri

Anumite comenzi LaTeX produc un text predefinit, de regulă titluri de unităţi secţionale. De exemplu, comanda `\tableofcontents` are ca efect scrierea cuprinsului, precedat de titlul "Table of Contents". Desigur că acest titlu este nepotrivit pentru o lucrare scrisă în altă limbă decât engleza. De aceea, a fost prevăzută posibilitatea de a modifica aceste titluri. Pentru exemplul nostru, în preambulul fişierului sursă trebuie folosită comanda `\renewcommand{\contentsname}{Cuprins}`. Aceasta nu are nici un efect imediat, dar dacă mai târziu se va folosi comanda `\tableofcontents`, cuprinsul generat va avea titlul "Cuprins". Secvenţa de control `\contentsname` este o comandă LaTeX care are implicit ca efect tipărirea şirului de caractere "Table of contents". Această comadă este apelată de comanda `\tableofcontents`.

Prin comanda `\renewcommand{\contentsname}{Cuprins}` dăm o nouă semnificaţie comenzii `\contentsname`, de aceea atunci când dăm comanda `\tableofcontents` această nouă semnificaţie va fi folosită, producând rezultatul descris mai sus.

Mai multe amănunte despre comanda `\renewcommand`, mai târziu.

Comenzile ale căror definiţii trebuie modificate pentru a le adapta la limba română sunt următoarele:

`\partname` numele dat unei părţi (numai stilul `book` şi `report` — implicit *Part*);

`\chaptername` numele dat capitolului (numai stilul `book` şi `report` — implicit *Chapter*);

`\refname` numele dat bibliografiei în cazul stilului `article` (implicit *References*);

`\bibname` numele dat bibliografiei în cazul stilurilor `book` şi `report` (implicit *Bibliography*);

`\appendixname` numele dat anexelor (implicit *Appendix*);

`\abstractname` numele dat rezumatului (implicit *Abstract*);

`\indexname` numele dat indexului (implicit *Index*);

`\contentsname` numele dat cuprinsului (implicit *Contents*);

`\figurename` numele dat figurilor (implicit *Figure*):

`\listfigurename` numele dat listei figurilor (implicit *List of Figures*);

`\tablename` numele dat tabelelor (implicit *Table*);

`\listtablename` numele dat listei tabelelor (implicit *List of Tables*);

Dăm mai jos comenzile `\renewcommand` folosite de noi pentru culegerea acestei cărţi:

```
\renewcommand{\contentsname}{Cuprins}
\renewcommand{\listfigurename}{Lista figurilor}
\renewcommand{\listtablename}{Lista tabelelor}
\renewcommand{\figurename}{Figura}
\renewcommand{\tablename}{Tabelul}
\renewcommand{\chaptername}{Capitolul}
\renewcommand{\bibname}{Bibliografie}
\renewcommand{\appendixname}{Anexa}
```

2.3.3 Despărţirea în silabe

LaTeX rupe texte în rânduri pentru a formá paragrafe. Acest proces se realizează după un algoritm complex, care uneori trebuie să despartă anumite cuvinte în silabe. Despărţirea în silabe se face pe baza unui set de şabloane, care sunt specificate la iniţializarea LaTeX-ului (de regulă atunci când se face instalarea). Un asemenea set de şabloane descrie regulile de despărţire în silabe pentru *o singură limbă*! LaTeX are posibilitatea să folosească mai multe seturi de şabloane, utilizatorul putând opta pentru unul din ele prin

comanda `\language` care are ca parametru un număr. Întotdeauna comanda `\language0` selectează setul pentru limba engleză. Majoritatea sistemelor nu au alte seturi de şabloane, iar dacă au, trebuie să verificaţi în ghidul local (undeva pe sistemul de calcul pe care îl utilizaţi trebuie să existe un fişier numit `local.dvi`) pentru fiecare limbă ce număr este asociat[1].

Şabloanele sunt în aşa fel concepute încât LaTeX să nu despartă greşit în silabe. Uneori însă el nu găseşte toate silabele. Pentru astfel de cuvinte, se pot introduce noi reguli cu ajutorul comenzilor `\-` şi `\hyphenation`. Comanda `\hyphenation` are, spre deosebire de comanda `\-`, efect global.

De exemplu comanda `\hyphenation{pro-ce-sor pro-ce-sa-re}` are ca efect despărţirea în silabe a acestor cuvinte peste tot în text unde este nevoie (regula folosită la despărţire este cea dată ca argument al comenzii). O altă modalitate de a indica despărţirea în silabe a unui cuvânt este de a insera comanda `\-` în punctele de despărţire ale cuvântului. Scrierea cuvântului `procesare` în forma `pro\-ce\-sa\-re` defineşte regula de despărţire a acelui cuvânt în acel loc unde apare în text, efectul fiind deci local. În general acest mod de despărţire în silabe se foloseşte atunci când într-un loc în text a apărut eroarea *Overfull* `xxx`*pt to wide ...* (vezi capitolul 10).

[1] Există şabloane pentru limbile: engleză, germană, *română*, cehă şi poloneză. Pentru limbile germană şi engleză şabloanele sunt conţinute în pachetul de distribuţie, pentru limba română şabloanele se găsesc în lucrarea [7], iar pentru celelalte limbi ele sunt disponibile via **anonymous ftp**.

Capitolul 3

Organizarea logică a documentului

3.1 Structura generală a fişierului de intrare

3.1.1 Comanda `\documentstyle`

Un fişier LaTeX trebuie să înceapă cu comanda `\documentstyle`, eventual precedată de comenzi TeX. Nu este însă recomandabilă introducerea de comenzi TeX într-un fişier LaTeX dacă nu este neapărat nevoie; dacă totuşi acest lucru nu se poate evita, atunci să se facă *după* comanda `\documentstyle`.

Comanda `\documentstyle` are următoarea sintaxă:

`\documentstyle[`*opt1,opt2,...*`]{`*doc_style*`}`

unde:

opt1,opt2,...	este lista de opţiuni (care este opţională)
doc_style	este stilul documentului;

Stiluri disponibile:

`book` va organiza documentul sub forma unei cărţi. Printre caracteristici sunt:

- Culegerea textului se face diferenţiat pentru pagini pare şi impare;
- Este organizată pe părţi, capitole, secţiuni, subsecţiuni şi subsubsecţiuni;
- Dacă nu se specifică altfel prin comanda `\pagestyle`, antetul fiecărei pagini pare conţine titlul capitolului curent, iar al fiecărei pagini impare, titlul secţiunii curente. Fiecare capitol începe la o pagină nouă impară.

`report` va organiza documentul sub forma unui raport. E asemănător cu `book`, dar culegerea textului nu se face diferenţiat funcţie de paritatea paginii, iar capitolele pot începe şi pe pagini pare.

`article` va organiza documentul sub forma unui articol. Acesta nu poate avea părţi şi nici capitole. Implicit antetele sunt lăsate goale, şi culegerea textului se face nediferenţiat după paritatea paginilor.

`letter` permite scrierea de scrisori oficiale. Acest stil de document este discutat mai pe larg în anexa D.

Opţiuni disponibile:

`11pt` Setează mărimea caracterelor normale la 11 puncte tipografice. Implicit, aceasta este de 10 puncte tipografice. Celelalte dimensiuni vor fi raportate la dimensiunea caracterelor normale (vezi tabelul **6.2** de la pagina 64).

`12pt` Idem ca mai sus, dar setează mărimea caracterelor normale la 12 puncte tipografice.

`fleqn` „Flush Left EQuatioN" – determină alinierea la stânga a formulelor matematice. Implicit, acestea sunt centrate pe pagină.

`german` Pune la dispoziţie comenzi utile pentru texte în limba germană .

`ifthen` Pune la dispoziţie instrucţiuni de test şi de ciclare.

`leqno` „Left EQuation no." Determină aşezarea în stânga a numerelor asociate ecuaţiilor. Implicit acestea sunt aşezate în dreapta ecuaţiilor.

`makeidx` Util la documentele în care se doreşte obţinerea (semi)automată a unui index. Amănunte în paragraful 4.12 (pagina 48).

`showidx` Oferă o comandă ce permite aşezarea ca note marginale a cuvintelor cheie care sunt trecute în index. Vezi paragraful 4.12 (pagina 48).

`twocolum` Determină culegerea textului pe două coloane.

`twoside` Determină aşezarea diferenţiată în pagină, pentru pagini pare şi impare. Este implicit pentru stilul `book`.

`toc_entry` În cazul în care există în cuprins secţiuni cu număr foarte lung, apar probleme. `toc_entry` pune la dispoziţie comenzile: `\adjustbook`, `\adjustreport` şi `\adjustarticle` pentru eliminarea acestor neajunsuri.

`titlepage` Dacă se doreşte o pagină de titlu la stilul `article`, unde implicit titlul e aşezat la începutul primei pagini de text. Este folosit numai împreună cu stilul `article`.

3.1.2 Comanda \begin{document}

Comanda `\begin{document}` este comanda de început de document. Între comenzile `\documentstyle` şi `\begin{document}` nu se poate insera text în document. Între aceste comenzi pot apărea doar alte comenzi TEX sau LATEX care nu au ca efect generare de text. Această zonă se numeşte *preambul*-ul fişierului. De exemplu, aici poate să apară comanda `\title`, care *setează* titlul documentului, dar nu poate să apară comanda `\maketitle` care pune titlul în pagină.

3.1.3 Comanda \end{document}

Comanda `\end{document}` marchează sfârşitul documentului. Toate caracterele ce urmează după această comandă vor fi ignorate. Tot textul ce urmează a fi cules trebuie să fie plasat între comenzile `\begin{document}` şi `\end{document}`.

După cum vom vedea în secţiunea 4.1, comenzile `\begin{document}` şi `\end{document}` delimitează un aşa-numit *cadru*. Cadrul `document` este cadrul obligatoriu pentru orice stil de document, el delimitându-l.

3.2 Construirea titlului documentului

Începând cu această secţiune vom face abstracţie de stilul `letter`, care va fi tratat separat în anexa D.

La începutul unui document apare în mod uzual titlul documentului, autorul — eventual cu adresă — şi eventual data elaborării acelui document. În LATEX, mai întâi se setează trei variabile corespunzătoare titlului, autorului şi datei. Pentru aceasta, există pentru cele trei stiluri de documente comenzile:

3.2.1 Comanda \title

Comanda `\title` setează titlul documentului. Dacă titlul nu încape într-un singur rând, LATEX *nu* va rupe rândul, deoarece un titlu nu poate fi împărţit pe rânduri după un algoritm, fiind importantă semnificaţia cuvintelor. De aceea, în acest caz titlul trebuie împărţit de către autor pe rânduri, prin introducerea comenzii `\\` acolo unde se doreşte trecerea la rând nou.

Această comandă trebuie să apară înainte de comanda `\maketitle`. Dacă comanda `\title` lipseşte, nu poate fi folosită nici comanda `\maketitle`.

`\title{`*text*`}`

3.2.2 Comanda \author

Comanda `\author` setează autorul documentului. De asemenea, dacă aici e nevoie de mai multe rânduri, trecerea la rând nou se face prin intermediul comenzii `\\`.

În cazul în care sunt mai mulţi autori ei vor fi separaţi pe rânduri cu ajutorul comenzii `\and`.

Această comandă poate lipsi, caz în care nu poate fi folosită comanda `\maketitle`. Dacă este folosită, comanda `\author` trebuie să apară înainte de `\maketitle`.

`\author{`*text*`}`

3.2.3 Comanda \date

Comanda `\date` setează data. Dacă nu se dă această comandă, comanda `\maketitle` va folosi data curentă de pe sistemul de calcul pe care se face procesarea.

`\date{`*text*`}`

3.2.4 Comanda \maketitle

Punerea în pagină a titlului, autorului şi a datei se face prin intermediul comenzii `\maketitle`, care trebuie să apară în mod obligatoriu *după* comanda `\begin{document}`, şi *după* comenzile `\title`, `\author` şi opţional `\date`.

Principial, comanda `\maketitle` poate să apară oriunde între comenzile `\begin{document}` şi `\end{document}`, dar dacă nu este prima comandă de după `\begin{document}`, este probabil ca rezultatul să nu fie cel dorit.

Comanda `\maketitle` are efecte diferite, în funcţie de stilul documentului. Astfel:

- Pentru `book` şi `report`, titlul, autorul şi data vor ocupa o pagină întreagă, pagina 0.
- Pentru `article`, aceste informaţii vor apărea la început de pagină, iar în continuare va începe textul documentului fără a se trece la pagină nouă.

Fontul este ales de LaTeX, diferit pentru titlu, autor şi dată, şi toate rândurile titlului vor fi centrate pe pagină.

3.2.5 Trimiteri la note de subsol

Oriunde în textul introdus ca parametru la comenzile `\title`, `\author`, `\date` şi comezile secţionale se pot insera trimiteri la note de subsol, cu ajutorul comenzii `\thanks`[1]. Se recomandă „lipirea" acestei comenzi la sfârşitul cuvântului la care se referă.

`\thanks{`*text*`}`

Este posibilă de asemenea o trimitere la o notă de subsol deja folosită cu ajutorul comenzii `\footnotemark`.

`\footnotemark[`*numărul notei*`]`

unde *numărul notei* este numărul de ordine al comenzii `\thanks` la al cărei text se face trimiterea.

Atenţie: `\thanks` este valabilă *numai şi numai* în argumentele comenzilor `\title`, `\author`, `\date` şi ale unităţilor secţionale! Pentru note de subsol obişnuite, vezi comanda `\footnote`.

Acesta este un titlu foarte lung.
Pot apărea şi trimiteri* la note de subsol.

Primul autor†
Al doilea autor †

21.10.1993

Începutul textului ...

*nota de titlu
†o adresa...

```
\documentstyle[11pt]{article}
\title{Acesta este un titlu foarte lung.\\
      Pot apărea \c{s}i trimiteri\thanks{
      nota de titlu}\\ la note de subsol.
      }
\author{Primul autor\thanks{o adresa...}\\
        Al doilea autor\footnotemark[2]
       }
\date{21.10.1993}
\begin{document}
\maketitle
\^Inceputul textului \cdots
```

3.3 Capitole şi secţiuni

Orice lucrare este în general organizată pe părţi, capitole, secţiuni, subsecţiuni şi subsubsecţiuni, iar unele documente au şi un rezumat. LaTeX ştie să

[1] Comanda `\thanks`, spre deosebire de `\footnote`, foloseşte alte semne pentru notele de subsol.

gestioneze aceste entităţi şi să facă aranjarea în pagină a lor, prin intermediul unor comenzi pe care le numim secţionale:

```
\part  \chapter  \section  \subsection  \subsubsection
```

Fiecare din aceste comenzi are trei forme. În forma simplă, comanda are un singur parametru delimitat de acolade. Acesta va fi un text care reprezintă titlul unităţii secţionale. Numerotarea părţilor se face automat. Totodată, dacă se generează un cuprins automat, acest text este folosit şi pentru cuprins, iar dacă stilul paginii este `heading`, ceea ce e implicit pentru stilul `book`, acelaşi text este folosit pentru antetul paginilor dacă este cazul.

Dacă nu se doreşte numerotarea unei unităţi secţionale, se va folosi forma „cu `*`“ a comenzii, adică:

```
\part*  \chapter*  \section*  \subsection*  \subsubsection*
```

În acest caz, unitatea secţională respectivă nu va fi trecută în cuprins, şi nici nu se va folosi pentru antetele paginilor.

În fine, mai există o formă, care de fapt este prima formă cu un parametru opţional care este transmis înaintea celui principal, şi este inclus în paranteze drepte, de exemplu pentru `\part`:

`\part[`*parametru_optional*`]{`*parametru_principal*`}`

Parametrul principal este folosit în text, iar cel opţional pentru cuprins şi pentru antet, dacă este cazul.

3.3.1 Comanda `\abstract`

Comanda `\abstract` se foloseşte împreună cu `\endabstract`, şi ele pot fi înlocuite cu cadrul `abstract` (vezi secţiunea 4.2).

```
\abstract
text...
...rezumat
\endabstract
```

Ele permit specificarea unui text care să fie tipărit într-un format corespunzător ca rezumat al lucrării, cu titlul „abstract“.

Comanda este valabilă numai pentru stilurile `report` şi `article`, pentru primul stil având ca efect aşezarea rezumatului pe o pagină separată.

Se recomandă introducerea acestei comenzi imediat după `\maketitle` sau după comenzile `\tableofcontents`, `\listoffigures` etc., dacă acestea sunt folosite la începutul documentului.

Pentru folosirea altui cuvânt decât „abstract" pentru titlul acestei secţiuni, vezi secţiunea 2.3.2.

3.3.2 Comanda \part

Comanda `\part` permite împărţirea unei cărţi sau unui raport pe părţi, ea fiind definită numai pentru stilurile `book` şi `report`.

`\part{`*titlul părţii*`}`
`\part*{`*titlul părţii*`}`
`\part[`*titlu scurt*`]{`*titlul părţii*`}`

Efectul comenzii depinde de stilul documentului în care este folosită.

La `book` are ca efect trecerea la o pagină nouă cu număr impar, pe care se aşază cuvântul „Part" urmat de numărul părţii şi titlul ei, centrate vertical pe pagină, şi saltul la următoarea pagină nouă cu număr impar.

La `report` are ca efect trecerea la o nouă pagină, necontând dacă este una pară sau impară, aşezarea numărului părţii şi a titlului ei centrat pe pagină şi din nou trecere la pagină nouă.

Pentru adaptarea comenzii la alte limbi, vezi secţiunea 2.3.2.

3.3.3 Comanda \chapter

Comanda `\chapter` este de asemenea rezervată stilurilor `book` şi `report`.

`\chapter{`*titlul capitolului*`}`
`\chapter*{`*titlul capitolului*`}`
`\chapter[`*titlu scurt*`]{`*titlul capitolului*`}`

Numerotarea capitolelor nu este afectată de comanda `\part`.

Efectul comenzii este trecerea la pagină nouă (la stilul `book` la una impară), şi aşezarea la începutul paginii a cuvântului „Chapter", a numărului şi titlului capitolului.

Pentru adaptarea comenzii la alte limbi, vezi secţiunea 2.3.2.

3.3.4 Comenzile \section, \subsection şi \subsubsection

Aceste trei comenzi secţionale permit trecerea la o secţiune, sub- sau sub-sub-secţiune nouă. Dăm doar cele trei forme pentru comanda `\section`:

`\section{`*titlul secţiunii*`}`
`\section*{`*titlul secţiunii*`}`
`\section[`*titlu scurt*`]{`*titlul secţiunii*`}`

Numerotarea acestor unităţi secţionale este afectată de comenzile `\chapter`, `\section`, `\subsection` sau `\subsubsection` date anterior, astfel încât să se asigure o notare firească a unităţilor secţionale.

Aceste trei comenzi sunt valabile pentru `book`, `report` şi `article`. La stilurile `book` şi `report`, secţiunile sunt notate cu două numere separate prin punct, primul număr reprezentând capitolul iar al doilea secţiunea din cadrul lui. La stilul `article` se notează cu un singur număr. Pentru notarea subsecţiunilor şi a subsubsecţiunilor se mai adaugă câte un (respectiv două) număr (numere).

De exemplu 3.1.2.4 va fi subsubsecţiunea 4 a subsecţiunii 2 a secţiunii 1 din capitolul 3 (dacă se foloseşte stilul de document `book` sau `report`).

3.3.5 Comanda `\appendix`

Această comandă modifică modul de numerotare a capitolelor şi a secţiunilor.

`\appendix`

De exemplu, în stilul `article`, secţiunile declarate după comanda `\appendix` vor fi numerotate cu „A“, „B“ etc.. În stilul `book`, după o comandă `\appendix` capitolele vor fi notate cu „Appendix A“, „Appendix B“ etc..

3.4 Etichete şi referiri

Adesea devine necesară referirea la unele capitole, secţiuni, ecuaţii, tabele, figuri etc.. Deoarece LaTeX numerotează singur toate aceste entităţi, nu putem şti în timpul editării numărul la care să ne referim.

De aceea, s-a prevăzut posibilitatea aplicării de etichete pe diferite entităţi, şi referirea la ele. Comenzile aferente sunt `\label` şi `\ref`.

`\label{`*eticheta*`}`
`\ref{`*eticheta*`}`

unde *eticheta* este un nume simbolic pentru acea entitate.

Deosebit de utilă este şi comanda `\pageref` care permite referirea la pagina pe care apare o entitate etichetată.

`\pageref{`*eticheta*`}`

Atenţie: Dacă se folosesc referiri la etichete, este nevoie de procesarea de două ori la rând a fişierului pentru rezolvarea referinţelor încrucişate, deoarece la prima procesare abia se scriu în fişierul `.aux` informaţii despre entităţile etichetate, urmând ca la a doua procesare informaţiile să fie preluate de aici pentru a fi folosite de comenzile `\ref` şi `\pageref`.

3.4.1 Etichetarea

```
\subsection{Etichetarea}\label{s.e.}
Comanda ...
```

Comanda ...

Comanda `\label{`*eticheta*`}` are ca efect scrierea în fişierul `.aux` a unei linii cu următorul conţinut:

`\newlabel{`*eticheta*`}{{ `*simbol*`}{`*pagina*`}}`.

Parametrul *eticheta* este numele simbolic ce apare ca parametru al comenzii `\label`, *simbol* este simbolul cadrului de lucru (secţiunea, tabelul, ecuaţia etc.) curent iar *pagina* este numărul de pagină pe care a apărut comanda `\label`. În acest exemplu, în fişierul `.aux` s-a scris:

`\newlabel{s.e.}{{3.4.1}{25}}`.

3.4.2 Referirea

```
\subsection{Referirea}
\^In paragraful \ref{s.e.} de la pagina
\pageref{s.e.} am discutat despre
etichetare. ...
```

În paragraful 3.4.1 de la pagina 25 am discutat despre etichetare.

După cum se vede din exemplul de mai sus, referirea este la fel de simplă: comanda `\ref{`*eticheta*`}` se poate folosi pentru obţinerea simbolului obiectului în cadrul căruia a fost plasată comanda `\label`, iar comanda `\pageref{`*eticheta*`}` pentru obţinerea paginii pe care a fost plasat obiectul respectiv.

Atenţie! O unitate secţională poate ocupa mai multe pagini. De aceea, este de dorit ca etichetarea să se facă imediat după comanda secţională.

3.5 Cuprinsul

LaTeX permite generarea automată a cuprinsului, a listei figurilor şi a listei tabelelor din lucrare, cu ajutorul comenzilor:

```
\tableofcontents
\listoffigures
\listoftables
```

Aceste comenzi pot apărea oriunde în text. Dacă se foloseşte una din aceste comenzi în document, acesta trebuie procesat de două ori consecutiv pentru obţinerea rezultatului dorit, sau chiar de trei ori, dacă una din listele generate sau cuprinsul depăşeşte două pagini în cazul stilului `book`, repectiv o pagină pentru celelalte stiluri.

Efectul acestor trei comenzi este includerea în locul în care apar a unui fişier ce are acelaşi nume cu fişierul rădăcină şi extensia `.toc` pentru `\tableofcontents`, `.lof` pentru `\listoffigures` şi `.lot` pentru `\listoftables`, precum şi scrierea respectiv actualizarea acestui fişier.
Scrierea oricăror fişiere în afară de `.dvi` şi `.log` poate fi inhibată prin comanda `\nofiles` plasată în preambul.

Intrările în fişierul `.toc` sunt scrise de comenzile secţionale, iar intările în fişierele `.lof` şi `.lot` sunt scrise de comenzile `\caption`.

Putem adăuga intrări în oricare din cele trei fişiere cu următoarele comenzi:

`\addcontentsline{`*fişier*`}{`*unitate secţională*`}{`*text*`}`

`\addtocontents{`*fişier*`}{`*text*`}`

unde:

fişier poate fi: `toc`, `lof` sau `lot`.

unitate secţională poate fi:

- Nume de secţiune (`part`, `chapter`, `section` etc.) dacă *fişier* este `toc`.
- `figure` dacă *fişier* este `lof`.
- `table` dacă *fişier* este `lot`.

text este text ordinar sau/şi comenzi LaTeX.

Capitolul 4

Cadruri

Termenul în limba engleză este *environment* (cadru). Un cadru delimitează o *entitate*, cum ar fi de exemplu o ecuaţie, un tabel, o figură, şi el este folosit pentru a aşeza în pagină un anumit text, după un format specific entităţii respective.

Un cadru este delimitat de comenzile `\begin` şi `\end`:

`\begin{`*nume_cadru*`}`
text
`\end{`*nume_cadru*`}`

unde *nume_cadru* este numele unui cadru, iar *text* este textul ce trebuie cules în formatul corespunzător acelui cadru, de exemplu pentru o ecuaţie simplă:

...se vede din ecuaţia:

$$x_n = x_{n-1} + x_{n-2} \qquad n \geq 2, \qquad (4.1)$$

nu se poate ...

```
\dots se vede din ecuaţia:
\begin{equation}
x_n = x_{n-1} + x_{n-2} \qquad n\ge 2,
\end{equation}
nu se poate ...
```

Cadrurile pot fi imbricate, dar trebuie ţinut seama că ultimul cadru deschis trebuie închis primul.

4.1 Documentul

Cadrul `document` este obligatoriu pentru orice fişier LaTeX. El este cadrul general în care sunt accesibile comenzile LaTeX care au ca efect scriere de text (inclusiv textul ordinar). Acest cadru nu poate să apară decât o singură

dată, sfârşitul lui reprezentând şi sfârşitul documentului. Nici o comandă TEX sau LATEX ce produce text nu este validă în afara acestui cadru.

Înainte de începutul cadrului `document`, trebuie să apară comanda `\documentstyle`, care stabileşte tipul documentului.

4.2 Rezumatul (Abstract)

Cadrul `abstract` este cel care permite aşezarea în pagină a rezumatului lucrării, aşa cum se obişnuieşte la articole. El este opţional pentru `report` şi `article`, dar nu este definit pentru `book` şi `letter`.

Acest cadru va apărea o singură dată în document, de obicei undeva la începutul lucrării, de regulă imediat după cuprins, dacă acesta există şi nu este la sfârşitul lucrării, sau, în caz contrar, imediat după titlu.

În locul acestui cadru se poate folosi şi perechea de comenzi `\abstract` şi `\endabstract` (vezi secţiunea 3.3.1).

4.3 Citate

Citatele trebuie aşezate cu mare atenţie în pagină, iar acest lucru trebuie făcut unitar într-o lucrare. LATEX pune la dispoziţie trei cadruri pentru citate, toate fiind de fapt variante de liste.

4.3.1 Cadrul `quote`

Cadrul `quote` este destinat citatelor scurte, care au un singur paragraf, pentru exemple sau alte texte care trebuiesc evidenţiate.

O regulă de bază a tipografilor:

> Cu cât sunt folosite mai puţine fonturi, cu atât textul este mai lizibil.

De aceea, să lăsăm în seama altora alegerea fonturilor pentru textele ce trebuiesc evidenţiate.

```
O regul\u{a} de baz\u{a} a tipografilor:
\begin{quote}
Cu c\^{a}t sunt folosite mai pu\c{t}ine
fonturi, cu at\^{a}t textul este mai
lizibil.
\end{quote}
De aceea, s\u{a} l\u{a}s\u{a}m \^{\i}n
seama altora alegerea fonturilor pentru
textele ce trebuiesc eviden\c{t}iate.
```

Citatul poate avea mai multe paragrafe, dar nu se lasă alineat la începutul lor, de aceea pentru astfel de citate, compuse din mai multe paragrafe, se recomandă cadrul `quotation`.

4.3.2 Cadrul `quotation`

Cadrul `quotation` este asemănător cu `quote`, dar permite evidenţierea unui text ce are mai multe paragrafe. Diferenţa majoră dintre `quote` şi `quotation` este că în ultimul cadru paragrafele încep cu un alineat.

... lucrare din care citäm în continuare:

> Pe cât este de adevărat faptul că rândurile prea lungi obosesc privirea, pe atât de greşită este ideea de a scurta rândurile şi a folosi text pe mai multe coloane atunci când nu este neapărat nevoie.
>
> La o analiză mai atentă a rândurilor scurte, se constată că apar probleme la alinierea textului în partea dreaptă, ceea ce duce la distanţe prea mari între cuvinte.

Acest lucru se poate vedea şi din aranjarea citatului de mai sus.

```
... lucrare din care cit\u{a}m \^{\i}n
continuare:
\begin{quotation}
Pe c\^{a}t este de adev\u{a}rat faptul c\u
{a} r\^{a}ndurile prea lungi obosesc
privirea, pe at\^{a}t de gre\c{s}it\u{a} e
ideea de a scurta r\^{a}ndurile \c{s}i a
folosi text pe mai multe coloane atunci
c\^{a}nd nu este neap\u{a}rat nevoie.\par
La o analiz\u{a} mai atent\u{a} a r\^
{a}ndurilor scurte, se constat\u{a} c\u{a}
apar probleme la alinierea textului \^\i n
partea dreapt\u a, ceea ce duce la
distan\c te prea mari \^{\i}ntre cuvinte.
\end{quotation}
Acest lucru se poate vedea \c{s}i din
aranjarea citatului de mai sus.
```

4.3.3 Cadrul `verse`

Cadrul `verse` se pretează cel mai bine pentru punerea în pagină a poeziilor. Versurile vor fi despărţite între ele cu ajutorul comenzii `\\` iar strofele prin rânduri libere:

A fost odată ca-n poveşti,
A fost ca niciodată, ...

Şi era una la părinţi
Şi mândră-n toate cele,

```
\begin{verse}
A fost odat\u a ca-n pove\c sti,\\
A fost ca niciodat\u a, ...

\c Si era una la p\u arin\c ti\\
\c Si m\^ andr\u a-n toate cele,\\
\end{verse}
```

4.4 Alinieri

Atunci când LaTeX procesează un text, el încearcă implicit să îl alinieze atât la stânga cât şi la dreapta, aşa ca în acest paragraf. Există însă situaţii

în care nu dorim acest lucru. În cadrul `flushleft` se face doar alinierea începuturilor de rând, astfel încât marginea din dreapta a textului rămâne neregulată. Cadrul `flushright` permite aşezarea textelor aliniate numai la dreapta (mai rar folosită). În fine, pentru texte centrate există cadrul `center`:

Un text aliniat doar la stânga se pune într-un cadru `flushleft`.

Un text aliniat doar la dreapta se pune într-un cadru `flushright`.

Un paragraf centrat se va pune într-un cadru `center`.

```
\begin{flushleft}
Un text aliniat doar la st\^{a}nga se ...
\end{flushleft}
\begin{flushright}
Un text aliniat doar la dreapta se ...
\end{flushright}
\begin{center}
Un paragraf centrat se va pune ...
\end{center}
```

4.5 Păstrarea nemodificată a textului

De multe ori este nevoie de introducerea în document a unui text în care apar caractere rezervate LaTeX, sau care din alte motive dorim să nu fie modificate nicicum de către LaTeX. Pentru acest lucru ne stă la dispoziţie comanda `\verb`:

`\verb`*ctt ... ttc* sau:
`\verb*`*ctt ... ttc*

unde *c* este un caracter ce nu este o literă iar secveţa *tt ... tt* o secvenţă de orice şi oricâte caractere diferite de *c*. Diferenţa dintre `\verb` şi `\verb*` este că cea de-a doua formă tipăreşte spaţiul ca pe un semn ␣.

Cu ajutorul ei se pot introduce scurte şiruri de caractere în text, cum ar fi de exemplu:

`\documentstyle` este comanda care trebuie să preceadă orice ... cum ar fi caracterul |. `linia noua` este interpretată ca un spaţiu real, nu ca ␣.

```
\verb|\documentstyle| este comanda care
trebuie s\u{a} precead\u{a} orice ... cum
ar fi caracterul \verb.|..\par
\verb*%linia
noua% este interpretat\u a cu un spa\c tiu
real, nu ca \verb*/ /.
```

Dacă se doreşte aşezarea în pagină 'ad literam', a mai mult de câteva cuvinte, se foloseşte cadrul `verbatim` sau `verbatim*`. Diferenţa dintre cele două este că acesta din urmă tipăreşte spaţiile ca ␣.

Acest cadru permite culegerea fără formatare a unui text oricât de lung cu singura condiţie ca în acest text să nu apară secvenţa de caractere `\end{verbatim}`. Pentru inserarea acestei comenzi se va folosi comanda

`\verb|\end{verbatim}|` – dar şi aceasta, evident, numai în afara cadrului `verbatim`.

Atenţie! Comanda `\verb` şi cadrul `verbatim` nu pot apărea ca argument al altor comenzi!

Peste tot în lucrarea de faţă, coloanele din partea dreaptă a exemplelor au fost culese cu ajutorul acestui cadru.

4.6 Liste

Listele sunt foarte des folosite la enumerări, definiţii etc. LaTeX pune la dispoziţie trei tipuri de liste predefinite, precum şi posibilitatea ca utilizatorul să-şi definească propriile liste.

4.6.1 Cadrul `itemize`

Cadrul `itemize` permite înşirarea mai multor texte, fiecare din ele fiind astfel formatat încât să se facă o distincţie clară între ele. Fiecare element al listei va fi marcat cu acelaşi semn, pe care îl vom numi etichetă.

Chibritul se împarte în trei părţi egale:

- Partea de sus a chibritului. Aceasta se împarte în trei părţi egale:
  - Partea de sus a părţii de sus a chibritului;
  - Partea de mijloc a părţii de sus a chibritului;
  - Partea de jos a părţii de sus a chibritului;
- Partea de mijloc a chibritului.
- Partea de jos a chibritului

```
Chibritul se \^\i mparte \^\i n trei p\u
ar\c ti egale:
\begin{itemize}
\item Partea de sus a chibritului. Aceasta
se \^\i mparte \^\i n trei p\u ar\c ti
egale:
\begin{itemize}
 \item Partea de sus a p\u ar\c tii de sus
 a chibritului;
 \item Partea de mijloc a p\u ar\c tii de
 sus a chibritului;
 \item Partea de jos a p\u ar\c tii de sus
 a chibritului;
\end{itemize}
\item Partea de mijloc a chibritului.
\item Partea de jos a chibritului
\end{itemize}
```

Dacă mai multe liste de tip `itemize` sunt imbricate, atunci LaTeX va alege pentru marcarea elementelor etichete diferite, în funcţie de lista de care aparţin, ca în exemplul de mai sus.

În exemplul de mai sus, s-au folosit etichetele implicite pentru evidenţierea elementelor. Este posibilă însă şi impunerea acestora prin intermediul parametrului opţional pe care îl acceptă secvenţa de control `\item`:

— Intrarea este...
— Ieşirea a fost...

```
\begin{itemize}
\item[$\longrightarrow$] Intrarea este...
\item[$\longleftarrow$] Ie\c sirea a
  fost...
\end{itemize}
```

4.6.2 Cadrul enumerate

Cadrul **enumerate** permite enumerarea unor elemente. Este asemănător cu **itemize**, cu excepţia faptului că pentru evidenţierea elementelor listei se folosesc numere sau litere ce se incrementează automat la trecerea la un nou element. Şi aceste liste pot fi imbricate.

Dacă se foloseşte **\item** cu parametru, comportarea acestui cadru este identică cu **itemize**.

Fiecare element primind un număr, el poate fi etichetat cu ajutorul comenzii **\label** pentru a putea fi referit. Exemplu:

... următorii paşi:

1. Se ia un ...
2. Se desface capacul ...

Pasul 2 e ...

```
    ... urm\u atorii pa\c si:
\begin{enumerate}
\item Se ia un ...
\item\label{desfacerea} Se desface capacul
 ...
\end{enumerate}
Pasul \ref{desfacerea} e ...
```

4.6.3 Cadrul description

Acest cadru este util pentru formatarea unor glosare, sau a unei liste de definiţii. În acest caz folosirea lui **\item** fără parametru nu are sens (deşi nu produce o eroare), deoarece tocmai parametrul lui **\item** este eticheta elementului din listă, el fiind cules într-un font deosebit de restul textului:

itemize este o listă.

enumerate este o listă ordonată.

description este o listă de cuvinte cu explicaţii.

```
\begin{description}
\item[itemize] este o list\u a.
\item[enumerate]este o list\u a ordonat...
\item[description] este o listă de
cuvinte cu explicaţii.
\end{description}
```

Toate listele prezentate pot fi amestecate între ele, cu condiţia, desigur, ca ultima deschisă să fie prima închisă.

4.6.4 Cadrul `list`

Adesea utilizatorul are nevoie de un tip mai special de liste, care nu pot fi obţinute cu cele trei cadruri prezentate. Pentru acestea se va folosi cadrul `list`, care a fost folosit de altfel şi în definirea celorlalte trei tipuri:

`\begin{list}{`*eticheta implicită*`}{`*declaraţii*`}`
lista de elemente
`\end{list}`

unde:

lista de elemente este textul cu elementele listei. Fiecare element va începe cu comanda `\item`.

eticheta implicită este eticheta folosită de comenzile `\item` fără parametru opţional.

declaraţii este o secvenţă de declaraţii pentru modificarea parametrilor de formatare. Aceşti parametri sunt prezentaţi în figura **4.1**. Toţi sunt lungimi ce pot fi exprimate în orice unităţi de măsură acceptate de LaTeX şi pot fi modificaţi cu comenzile prezentate în secţiunea 9.1.

În afara acestor lungimi, următoarele comenzi controlează formatarea:

`\makelabel{`*label*`}` este comanda care poziţionează eticheta transmisă ca parametru opţional al comenzii `\item`. Poate fi redefinită de utilizator cu ajutorul comenzii `\renewcommand` în secţiunea de declaraţii.

`\usecounter{`*contor*`}` pune la dispoziţie contorul *contor* (care trebuie să fie creat!) (vezi secţiunea 9.2) pentru numerotarea internă a elementelor. Acest contor este automat iniţializat cu zero, iar la fiecare comandă `\item` fără parametru el va fi incrementat cu `\refstepcounter`, valoarea lui devenind valoarea curentă de referire (vezi `\label`).

Iată un exemplu simplu:

Acesta este textul care precede lista. El e aliniat la stânga şi la dreapta.

Obs.*1* Acesta este primul element din listă. Observaţi marginile din dreapta şi din stânga. Ambele sunt 5mm.

Obs.*2* Acesta este cel de-al doilea element. Observaţi ce se întâmplă cu eticheta itemului dacă nu are loc în cei 5mm puşi la dispoziţie.

Acesta este textul ce urmează listei. Este de asemenea aliniat la ambele margini.

```
\newcounter{obs}
 ...................
Acesta este textul care precede lista. ...
\begin{list}{{\bf Obs.\it\arabic{obs}}}
{
 \usecounter{obs}
 \setlength{\rightmargin}{5mm}
 \setlength{\leftmargin}{\rightmargin}
 \setlength{\labelwidth}{5mm}
}
\item Acesta este primul element din ...
\item Acesta este cel de-al doilea ...
\end{list}
Acesta este textul ce urmeaz\u a listei...
```

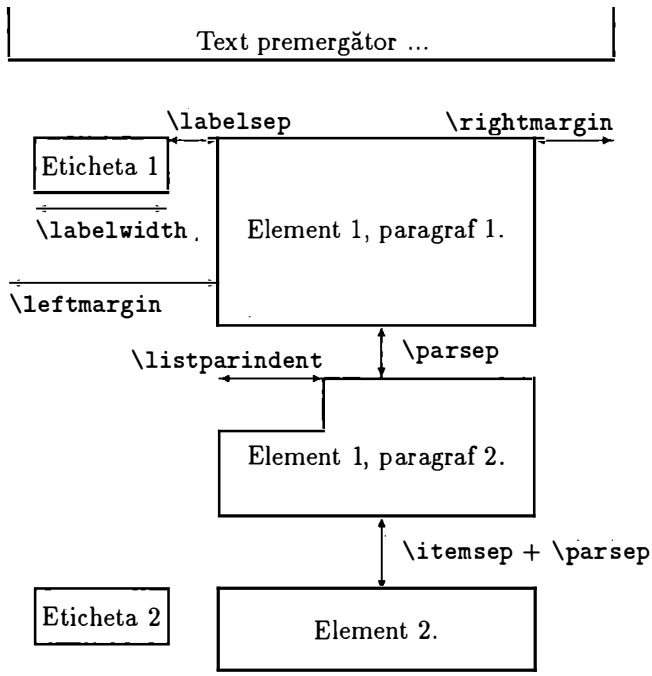


Figura 4.1: Semnificaţia parametrilor ce controlează formatul unui cadru de tip `list`.

4.7 Tabele

LaTeX pune la dispoziţie două cadruri pentru culegerea tabelelor:

4.7.1 Cadrul `tabbing`

Cadrul `tabbing` este destinat cu precădere alinierii textelor pe mai multe coloane. De fapt nu ne oferă decât un mecanism destul de primitiv pentru setarea şi folosirea unor tabulatori. În acest cadru, ne stau la dispoziţie următoarele secvenţe de control:

`\=`	are ca efect setarea unei poziţii de **tab**.
`\kill`	nu tipăreşte rândul curent.
`\>`	produce un salt orizontal la următorul **tab**.

`\<`	are efect opus comenzii `\>`, producând un salt orizontal la **tab**-ul precedent.
`\\`	produce trecerea la rând nou.
`\pushtabs`	salvează poziţia curentă a tuturor tabulatorilor care, ulterior, vor fi restauraţi cu comanda `\poptabs`.
`\poptabs`	vezi `\pushtabs`. Comenzile `\pushtabs` şi `\poptabs` pot fi imbricate dar, ele trebuie să apară întotdeauna în perechi într-un cadru dat.
`\a=, \a',\a‘`	produc accentele pe care în mod normal le-ar produce comenzile `\=,\'` şi `\‘`, aici redefinite.
`\+`	are ca efect începerea rândului următor de la următorul **tab**. Mai multe comenzi `\+` au efect cumulativ. Iniţial valoarea tabulatorului marginii stângi este setat la 0.
`\-`	are ca efect descreşterea valorii tabulatorului marginii stângi (care trebuie să fie pozitiv) cu unu, anulând astfel efectul unei comenzi `\+` începând cu următorul rând.
`\'`	are ca efect trecerea textului din coloana curentă situat înaintea comenzii în coloana precedentă, unde va fi aliniat la dreapta.
`\‘`	produce alinierea la dreapta tabelului a textului ce urmează comanda. După această comandă nu se vor mai da comenzile `\>`, `\=`, `\'` sau `\‘`.

Nr. crt	Nume şi	Prenume
1.	Icsulescu	Igrecel
2.		Señor Carlos Jose
3.	Constantin	Mitru

```
\begin{tabbing}
Nr. crt\quad \= Constantin\quad \= \kill\\
Nr. crt \> Nume \> \c si \'Prenume \\
1.      \> Icsulescu  \> Igrecel \\
2.      \> \‘Se\a=nor Carlos \+\+\\
 Jose\-\-\\
3.      \> Constantin \> Mitru\\
\end{tabbing}
```

Cel mai simplu este să alegem cele mai lungi texte de pe fiecare coloană, şi să le punem, separate prin `\=`, pe o linie model, care nu se tipăreşte ci este folosită doar pentru setarea tabulatorilor.

4.7.2 Cadrul `tabular`

Cadrul `tabular` se pretează foarte bine la construcţia de tabele fiind mai puternic decât cadrul `tabbing`.

Sintaxa generală a acestui cadru este:

`\begin{tabular}{`*format*`}` *rânduri* `\end{tabular}`
`\begin{tabular}[`*poziţie*`]{`*format*`}` *rânduri* `\end{tabular}`
`\begin{tabular*}{`*lăţime*`}[`*poziţie*`]{`*format*`}` *rânduri* `\end{tabular*}`

Parametrii au următoarea semnificaţie:

- *lăţime* specifică lăţimea cadrului `tabular`.
- *poziţie* specifică poziţia pe verticală faţă de punctul curent (implicit este centrat). Caracterele posibile sunt `t` şi `b` corespunzătoare plasării liniei de început respectiv de sfârşit al tabelului în dreptul punctului curent.
- *format* specifică formatul pe coloane. Este o secvenţă formată din specificatorii următori:
  - `l` specifică o coloană cu elemente aliniate la stânga;
  - `r` specifică o coloană cu elemente aliniate la dreapta;
  - `c` specifică o coloană cu elemente centrate;
  - `p{`*lat*`}` specifică o coloană de lăţime *lat* care conţine text aliniat la ambele margini.
  - `|` separator de coloane: linie verticală mărginită de spaţii.
  - `@{`*text*`}` separator de coloane: specifică o aşa numită *@-expresie*. Textul *text* va fi inserat în fiecare rând între coloanele între care apare *@-expresia*.
  - `*{`*număr*`}{`*coloane*`}` este echivalent cu *număr* de copii ale coloanelor, unde *număr* este un întreg pozitiv, iar *coloane* este o listă de specificatori de coloane;
- *rânduri* este o secvenţă de rânduri separate prin `\\`, fiecare rând fiind format din mai multe coloane separate prin `&`.

Pe lângă textul ordinar, ca element al unei coloane, pot să apară următoarele comenzi:

- `\multicolumn{`*număr*`}{`*format*`}{`*text*`}` textul *text* se va extinde peste *număr* coloane, fiind aliniat în conformitate cu caracterele din *format*.
- `\vline` produce o linie verticală de înălţime egală cu cea a rândului pe care apare.

- `\hline` produce o linie orizontală pe toată lăţimea tabelului.
- `\cline{`*i*`-`*j*`}` produce o linie orizontală întinzându-se de la coloana *i* până la coloana *j*.

Exemplu:

Denumirea		QTY	Preţ(mii lei)	
			unitar	total
Roşii		3 kg	0,3	0,9
Carne	Cal. I	2 kg	4	8
	Cal. II	3 kg	2,5	7,5

```
\begin{tabular}{*{2}{|c}|c|r|r|}
\hline\hline
\multicolumn{2}{|c|}{Denumirea}&
QTY&\multicolumn{2}{c|}{Preţ(mii lei)}\\
\cline{4-5}
\multicolumn{2}{|c|}{ }&&unitar&total\\
\hline\hline
\multicolumn{2}{|c|}{Roşii}&3 kg&0,3&0,9\\
\hline
Carne&Cal. I&2 kg&4&8\\
\cline{2-5}
 &Cal. II&3 kg&2,5&7,5\\
\hline\hline
\end{tabular}
```

4.8 Ecuaţii

Unul din marile avantaje pe care le oferă TEX este culegerea semnelor matematice. LATEX pune la dispoziţie o serie de cadruri pentru aşezarea acestora în pagină. În interiorul acestor cadruri se pot folosi toate comenzile modului matematic.

Aşezarea formulelor în pagină depinde de stilul folosit pentru elaborarea documentului (vezi `fleqn, leqno`). Implicit ecuaţiile sunt centrate orizontal iar numerele (pentru cele care primesc un număr) sunt aliniate la dreapta.

4.8.1 Cadrul `displaymath`

Cadrul `displaymath` permite aşezarea pe un singur rând a unei formule matematice care nu trebuie numerotată (pentru comparaţie, vezi `equation`).

unde a este dat de:

$$a = \frac{b^2 - c^2}{b + c}$$

sau:

$$a = \frac{b^2 - c^2}{b + c}$$

```
unde $a$ este dat de:
\begin{displaymath}
     a = \frac{b ^ 2 - c ^ 2}{ b + c }
\end{displaymath}
sau:
  $$ a = \frac{b ^ 2 - c ^ 2}{ b + c } $$
```

Acelaşi efect îl are şi introducerea ecuaţiei între o pereche de `$$`. Dar `$$` este o secvenţă TEX şi nu se recomandă folosirea ei în documente LATEX. Cadrul `displaymath` a fost introdus în LATEX pentru consecvenţă în ceea ce priveşte cadrurile.

4.8.2 Cadrul `equation`

Acest cadru permite pe lângă aşezarea unei ecuaţii în pagină şi numerotarea ei. Ecuaţia va primi un număr automat, iar acesta va fi aşezat împreună cu ecuaţia, în conformitate cu stilul ales (în mod normal, la dreapta ei).

Etichetarea se face imediat după începutul cadrului, folosind comanda `\label`.

Contorul folosit pentru gestionarea acestui număr se numeşte `equation` şi el poate fi controlat cu ajutorul comenzilor de gestiune a contoarelor.

$$x_n = x_{n-1} + 1 \qquad n > 1 \tag{4.2}$$

$$x_{n+1} = x_n + 1 \qquad n > 0 \tag{4.3}$$

Ecuaţiile 4.2 şi 4.3 sunt echivalente.

```
\begin{equation}\label{sirn}
x_n=x_{n-1}+1 \qquad n>1
\end{equation}
\begin{equation}\label{sirn+1}
x_{n+1}=x_n+1 \qquad n>0
\end{equation}
Ecuaţiile \ref{sirn} şi \ref{sirn+1}
sunt echivalente.
```

4.8.3 Cadrul `array`

Cadrul `array` se poate folosi oriunde e permisă folosirea unui simbol matematic (deci nu şi în mod text!). El e folosit la culegerea tabelelor alcătuite din simboluri matematice, fiind asemănător cu cadrul `tabular` care este folosit pentru tabele în mod text.

Formatul general este:

`\begin{array}{`*format*`}` *rânduri* `\end{array}`
`\begin{array}[`*poziţie*`]{`*format*`}` *rânduri* `\end{array}`
`\begin{array*}{`*lăţime*`}[`*poziţie*`]{`*format*`}` *rânduri* `\end{array*}`

unde parametrii au aceeaşi semnificaţie ca la cadrul `tabular`. Coloanele descrise prin `c`, `l`, `r` vor fi culese în mod matematic, iar coloanele `p{`*dim*`}` în mod text.

Exemplu de utilizare:

$$A = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & \cdots \\ \vdots & \vdots & \end{pmatrix}$$

```
\begin{displaymath}
A= \left( \begin{array}{ccc}
x_{11} & x_{12} & x_{13} \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array}\right)
\end{displaymath}
```

4.8.4 Cadrul eqnarray

Adesea este nevoie de scrierea succesivă a mai multor ecuaţii ce trebuiesc aliniate după un anumit criteriu (de exemplu, toate semnele = să apară unul sub celălalt).

Cadrul **eqnarray** conţine trei coloane. Trecerea la următoarea coloană se face cu **&**, iar trecerea la altă linie cu ****. Fiecare linie va fi numerotată ca o ecuaţie distinctă, dar poate fi folosită comanda **\nonumber** oriunde pe o linie pentru a inhiba numerotarea ei.

De asemenea, etichetarea se face, evident, tot pe linii.

$$f(x) = x^2 - 3 \tag{4.4}$$
$$g(x) = \int_0^{\pi} \sin(x)\, dx \tag{4.5}$$
$$h(x) = \sum_{i=1}^{n} x\frac{1}{i^2}$$

```
\begin{eqnarray}
f(x) & = & x^2-3 \\
g(x) & = & \int_0^{\pi} \sin(x) \, dx \\
h(x) & = & \sum \limits_{i=1}^nx\frac{1}
           {i^2}\nonumber
\end{eqnarray}
```

Dacă o linie trebuie aşezată fără a ţine cont de aliniere, se poate folosi comanda **\lefteqn**, al cărei argument va fi ecuaţia ce nu trebuie aliniată:

$$x_n = 1 + \frac{1}{2} + \\ + \frac{1}{4} \cdots + \frac{1}{2n} \tag{4.6}$$

```
\begin{eqnarray}
\lefteqn{x_n = 1+\frac{1}{2}+}\nonumber\\
 & & + \frac{1}{4} \cdots + \frac{1}{2n}
\end{eqnarray}
```

În toate cadrurile destinate aşezării ecuaţiilor în pagină, se pot folosi, desigur, toate semnele matematice puse la dispoziţie de TEX, care sunt prezentate în capitolul 7.

4.9 Desene

LaTeX pune la dispoziţie cadrul `picture` pentru realizarea unor desene simple:

`\begin{picture}`(*dim_x,dim_y*)(*start_x,start_y*)
desen
`\end{picture}`

Comanda are ca efect plasarea desenului reprezentat prin *desen* în punctul de coordonate *start_x ,start_y* (reamintim că poziţia curentă are coordonatele (0,0)), desenul are dimensiunile *dim_x,dim_y*. Dacă coordonatele punctului de start (care de fapt reprezintă offsetul faţă de punctul curent) lispesc atunci lor li se atribuie valori implicite nule. Unitatea de măsură folosită va fi cea specificată de comanda `\unitlength`.

Grosimea liniei cu care se desenează poate fi selectată cu comenzile:

- `\thinlines` selectează peniţa subţire;
- `\thicklines` selectează peniţa groasă;
- `\linethickness{`*gros*`}` selectează o peniţă de grosime *gros*;

desen-ul este în general alcătuit cu ajutorul comenzilor:

- `\put(`*coord_x,coord_y*`){`*obiect*`}` pune obiectul *obiect* în punctul de coordonate *coord_x, coord_y*;
- `\multiput(`*coord_x,coord_y*`)(`*dx,dy*`){`*număr*`}{`*obiect*`}` pune de *număr* de ori obiectul *obiect* începând cu punctul de coordonate *coord_x, coord_y*, de fiecare dată coordonatele vor fi incrementate cu *dx* respectiv *dy* (copia cu numărul *i* va fi pusă în punctul de coordonate (*coord_x+(i-1) * dx,coord_y+(i-1) * dy*);

Obiectele reprezentate de *obiect* pot fi text ordinar sau desene realizate cu comenzile:

`\makebox(`*dim_x, dim_y*`)[`*pozxy*`]{`*text*`}` are ca efect crearea unui cadru de lăţime *dim_x* şi înălţime *dim_y* punând în el textul *text* în poziţia specificată de parametrul opţional *pozxy* (implicit este centrat pe orizontală şi verticală). Parametrul *pozxy* este constituit din unul sau două caractere dintre caracterele `r,l,c,b,t` a căror semnificaţie este deja cunoscută.

`\framebox(`*dim_x,dim_y*`)[`*pozxy*`]{`*text*`}` comanda este similară cu comanda `\makebox` ca semnificaţie a parametrilor, dar de această dată se desenează şi marginea cadrului.

`\dashbox{`*dashdim*`}(`*dim_x,dim_y*`)[`*pozxy*`]{`*text*`}` comanda este analoagă cu comanda `\framebox`, dar de această dată marginea cadrului este desenată cu linie întreruptă. Distanţa dintre două linioare cât şi lungimea unei linioare fiind dată de valoarea lui *dashdim*.

`\frame{`*desen*`}` înconjoară desenul *desen* cu un cadru dreptunghiular tangent la acesta.

`\line(`x,y`){`*lung*`}` desenează o linie de lungime orizontală $x * lung$ şi lungime verticală $y * lung$. x şi y pot lua numai valori întregi prime între ele cuprinse în intervalul [-6,6]. Valoarea $y = 0$ înseamnă linie orizontală, iar valoarea $x = 0$ linie verticală indiferent de valoarea nenulă a celuilalt parametru.

`\vector(`x,y`){`*lung*`}` desenează un vector de lungime orizontală $x * lung$ şi lungime verticală $y * lung$. x şi y pot lua numai valori întregi prime între ele cuprinse în intervalul [-4,4]. Valoarea $y = 0$ înseamnă linie orizontală, iar valoarea $x = 0$ linie verticală indiferent de valoarea nenulă a celuilalt parametru.

`\shortstack[`*poz*`]{`*text*`}` creează o stivă de obiecte. Obiectele sunt aliniate conform lui *poz* (care poate lua valorile `l` sau `r`). Implicit obiectele vor fi puse centrat. *text* este o structură de tipul *obiect1* `\\` *obiect2* `\\` ... ultimul obiect fiind pus la baza stivei.

`\circle{`*d*`}` desenează un cerc de diametru *d* în punctul curent. Valoarea maximă a lui *d* depinde de implementare şi este în jur de 39 pt (depăşirea ei duce la desenarea unui cerc cu diametrul egal cu valoarea maximă admisă).

`\circle*{`*r*`}` desenează un disc de rază *r* în punctul curent. Valoarea maximă a lui *r* depinde de implementare şi este în jur de 15 pt (depăşirea ei duce la desenarea unui disc cu raza egală cu valoarea maximă admisă).

`\oval(`*dim_x,dim_y*`)[`*part*`]` desenează un oval (dacă *part* lipseşte) sau o parte din el. Ovalul va avea lăţimea dată de *dim_x* şi înălţimea dată de

dim_y. part poate fi o combinaţie de 1 sau 2 caractere dintre caracterele următoare: `l` (stânga), `r` (dreapta), `b` (partea de jos), `t` (partea de sus). De exemplu `br` înseamnă colţul din dreapta jos, iar `t` jumătatea superioară.

Toate aceste comenzi sunt mari consumatoare de memorie.

Iată mai jos sursa pentru desenul din figura **4.2**:

```
\begin{picture}(100,100) \setlength{\unitlength}{1pt}
\put(0,10){\makebox(40,15){test}} \put(0,30){\framebox(40,15){test}}
\put(0,50){\dashbox(40,15){test}}
\thicklines \put(60,10){\makebox(40,15){test}}
\put(60,30){\framebox(40,15){test}}
\put(60,50){\dashbox(40,15){test}} \thinlines
% deseneaza un patrat cu latura 104pt
\multiput(-2,-2)(104,0){2}{\line(0,1){104}}
\multiput(-2,-2)(0,104){2}{\line(1,0){104}}
% linie punctata cu \multiput
\multiput(-2,47)(2,0){52}{.}
% vectori cu \multiput
\multiput(-2,70)(20,0){5}{\shortstack[r]{$\vec{a}$ \\
\vector(1,2){10}}}
\put(51,12){\circle*{10}} \put(51,51){\circle{38}}
% centrul cercului ...
\put(50,51){\line(1,0){2}} \put(51,50){\line(0,1){2}}
\put(40,20){\frame{test}} \put(20,5){\oval(30,10)[t]}
\put(80,5){\oval(30,10)[b]} \put(50,5){\oval(20,10)[brtl]}
\put(-22,-14){\makebox(40,12){{\tiny (-2,-2)}}}
\put(104,104){\makebox(40,12){{\tiny (102,102)}}}
\end{picture}
```

Obiectele create cu comenzile `\makebox`, `\framebox` şi `\dashbox` pot fi salvate şi refolosite (procedeu descris pe larg în capitolul 8). Salvarea se face cu secvenţa de comenzi:

`\newsavebox{`*cda*`}`

`\savebox(`*dim_x, dim_y*`)[`*pozxy*`]{`*text*`}`.

Tipărirea cadrului cda salvat se face cu `\usebox{`*cda*`}`.

4.10 Cadruri mobile

LaTeX este destinat cu precădere scrierii de lucrări şiinţifice, în care se lucrează relativ des cu tabele şi figuri. Acestea nu pot fi puse întotdeauna

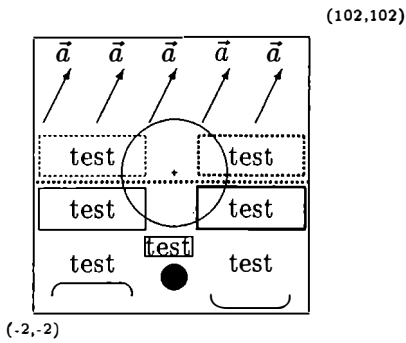


Figura 4.2: Exemplu cu toate comenzile LaTeX care pot fi transmise ca parametrii ai comenzilor \put şi \multiput

oriunde în text. Autorul poate lăsa pe seama lui LaTeX alegerea locului de amplasare, specificând figura sau tabelul într-un cadru mobil. El poate da însă ordinea de preferinţe pentru amplasarea cadrului mobil.

4.10.1 Cadrul `table`

Acest cadru este folosit pentru crearea de tabele. Formatul general al acestui cadru este:

```
\begin{table}[loc]
conţinutul tabelului
\end{table}

\begin{table*}[loc]
conţinutul tabelului
\end{table*}
```

Semnificaţia parametrilor este următoarea:

- *loc* este un parametru opţional şi specifică locul de amplasare a tabelului pe pagină şi poate fi o combinaţie a următoarelor caractere:
  - `h` reprezintă punctul curent (*here*);
  - `b` la baza unei pagini (*bottom*);
  - `t` la începutul unei pagini (*top*);

  - `p` pe o pagină separată, care conţine doar cadruri mobile, dar ţinând cont de anumite reguli, dintre care cea mai importantă este: niciodată înainte de locul în care se face pentru prima dată referire la tabel.

- *conţinutul tabelului* este constituit din comenzi LaTeX (sau/şi TeX) împreună cu text ordinar. De obicei este constituit dintr-un cadru `tabular`.

Forma cu * produce tabele ce se întind pe toată lăţimea paginii, chiar şi când se foloseşte opţiunea `twocolum`. Dacă nu se foloseşte această opţiune, între cele două forme nu există nici o diferenţă.

În conţinutul tabelului poate apărea comanda:

`\caption{` *text explicativ* `}`

care are ca efect numerotarea tabelului şi scrierea textului *text explicativ* ca notă explicativă. Implicit, numărul tableului este precedat de cuvântul "Table". Pentru schimbarea acestuia, vezi secţiunea 2.3.2. Comanda `\caption` se foloseşte de regulă imediat după `\begin{table}` sau imediat inainte de `\end{table}`, după cum dorim scrierea numărului şi a notei explicative deasupra sau sub conţinutul tabelului.

Comada `\caption` poate fi etichetată cu comanda `\label` pentru a se putea face referiri la tabel.

Comanda `\caption` mai poate primi un parametru suplimentar:

`\caption[`*intrare*`]{`*text*`}`

astfel că dacă se generează o listă de tabele sau de figuri cu una din comenzile `\listoftable` sau `\listoffigure` (după cum comanda aparţine unui cadru `table` sau unui cadru `figure`) în listă va fi folosit *intrare* şi nu *text*.

4.10.2 Cadrul `figure`

Acest cadru este folosit pentru crearea de figuri. Formatul general al acestui cadru este:

```
\begin{figure}[loc]
corpul figurii
\end{figure}

\begin{figure*}[loc]
corpul figurii
\end{figure*}
```

Semnificaţia parametrilor este aceeaşi cu cea a parametrilor cadrului `tabular`. De obicei *corpul figurii* este constituit dintr-un cadru `picture` centrat.

Iată mai jos un fragment din sursa acestei cărţi, prin care am specificat tabelul **4.1** şi figura **4.3**:

$a = 10$	$c = 15$
$b = 11$	$d = 0,15$

Tabelul 4.1: Valorile lui a,b,c,d

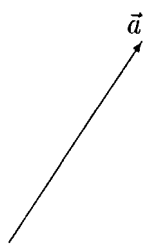


Figura 4.3: Vectorul $\vec{a}$ ...

```
Iată mai jos un fragment din sursa acestei cărţi, prin care am
specificat tabelul \ref{tab:abcd} şi figura \ref{fig:a}:

\begin{table}[htbp]
   \begin{center}
      \begin{tabular}{cc}
         $a=10$&$c=15$\\
         $b=11$&$d=0,15$
      \end{tabular}
   \end{center}
   \caption[{\bf Un mic exemplu}]{{\bf Valorile lui a,b,c,d}}
   \label{tab:abcd}
\end{table}

\begin{figure}[htbp]
   \begin{center}
     \begin{picture}(60,60)
        \put(0,0){\shortstack[r]{$\vec{a}$\\ \vector(2,3){50}}}
```

```
    \end{picture}
  \end{center}
  \caption{{\bf Vectorul $\vec{a}$ ...}}\label{fig:a}
\end{figure}
```

4.10.3 Pagini miniaturale

Pentru a culege un text de lăţime dictată de noi putem folosi comanda

`\parbox[`*loc*`]{`*lat*`}{`*text*`}`.

Aceasta are ca efect culegerea textului *text* într-o cutie de lăţime *lat*. Textul va fi procesat în mod paragraf, dimensiunea verticală a lui depinzând de numărul de linii rezultate în urma procesării. Alinierea verticală a cutiei faţă de rândul curent se face în funcţie de caracterul *loc* (`b`,`t`,`c`), care implicit este `c` (centrat).

În mod normal valoarea lui `\parindent` este zero (deci nu există alineat). Dacă se doreşte totuşi alineat, atunci va trebui setat `\parindent` la valoarea dorită cu comanda `\setlength`.

Pentru acelaşi scop, LaTeX pune la dispoziţie şi un cadru:

`\begin{minipage}[`*loc*`]{`*lat*`}` *text* `\end{minipage}`

Produce un cadru de lăţime *lat* dispus vertical faţă de rândul curent în funcţie de valoarea lui *loc* (`t`,`c`,`b`)— implicit este `c` (centrat). Spre deosebire de `\parbox` în `minipage` pot să apară ca *text* şi alte cadruri, ca `tabular` şi cadrurile care creează liste (nu se pot folosi însă cadruri mobile: `table` sau `figure`). O comandă `\footnote` sau `\footnotetext` produce o notă de subsol la sfârşitul cadrului `minipage` care uneori poate să fie plasată eronat atunci când cadrurile `minipage` sunt imbricate. Comanda `\footnote` nu se poate folosi în comanda `\parbox`.

Un cadru minipage care va începe cu o ecuaţie sau cu cadrul `eqnarray` poate avea la început un spaţiu suplimentar care poate fi înlăturat cu comanda `\vspace{-\abovedisplayskip}` plasată chiar la începutul cadrului `minipage`

4.11 Bibliografia

Majoritatea lucrărilor ştiinţifice au la sfârşit o listă cu lucrări la care s-a făcut referire în text, sau o listă de lucrări recomandate pentru a fi citite în legătură cu tema lucrării.

Lista bibliografică este produsă de cadrul `thebibliography` având următoarea sintaxă:

`\begin{thebibliography}{`*etichetă*`}`
lista de titluri
`\end{thebibliography}`

unde:

etichetă reprezintă cea mai largă etichetă care se va găsi în lista rezultantă. De exemplu, dacă sunt 10 lucrări şi se doreşte numerotarea lor, atunci *etichetă* trebuie să fie orice combinaţie de două cifre, deoarece cea mai largă etichetă se reprezintă pe două cifre.

lista de titluri reprezintă o listă cu intrări, fiecare începând cu comanda:

`\bibitem[`*etichetă_curentă*`]{`*cheie_de_citare*`}`

urmată de numele autorului, titlul cărţii etc.

Parametrul opţional *etichetă_curentă* este eticheta care se va folosi în listă în cazul în care nu se doreşte numerotarea lucrărilor. *cheie_de_citare* reprezintă cheia cu ajutorul căreia ne referim în text la o anumită lucrare din bibliografie (reprezintă parametrul care se transmite comenzii `\cite`).

Cadrul `thebibliography` poate apărea oriunde în document, dar de obicei este plasat la sfârşitul lui.

Bibliografia va începe pe o pagină nouă la stilurile `book` şi `report`, unde va avea titlul **Bibliography**, în timp ce la stilul `article` va purta numele **References**. Pentru schimbarea acestor titluri, vezi secţiunea 2.3.2.

Când vom face referiri în textul lucrării la o lucrare din bibliografie, o facem cu ajutorul comenzii `\cite`, a cărei sintaxă generală este:

`\cite[`*text*`]{`*listă_de_chei*`}`

unde *text* reprezintă un parametru opţional şi (dacă există) va fi adăugat ca o remarcă la citare. *listă_de_key* reprezintă o listă de una sau mai multe *chei_de_citare* separate prin virgulă.

Atenţie! Dacă se folosesc comenzi `\cite`, este obligatorie procesarea de două ori a fişierului sursă pentru ca referirile să fie corecte.

`\nocite{`*listă_de_chei*`}` nu produce nici o citare dar scrie *lista_de_chei* in fişierul `aux`. Comanda poate să apară oriunde în text după comanda `\begin{document}`.

Fiecare comandă `\bibitem` are ca efect scrierea în fişierul `.aux` a unei linii de genul

`\bibcite{`*cheie_de_citare*`}{`*număr*`}`

Aceste linii vor fi citite abia la a doua procesare a documentului. La prima procesare, comenzile `\cite` vor produce avertismentul de referinţă necunoscută. Dacă şi la a doua procesare avertismentul rămâne, înseamnă că undeva s-a strecurat o greşeală.

4.12 Index şi glosar

Cadrul `theindex` creează un index pe două coloane, fiecare intrare din acest index începând cu o comandă `\item`, `\subitem` respectiv `\subsubitem` şi are următoarea sintaxă:

`\begin{theindex}` *listă* `\end{theindex}`.

De regulă indexul se foloseţe pentru a indica paginile la care apar anumiţi termeni. El primeşte titlul **Index**, dar poate fi modificat prin redefinirea comenzii `\indexname` (vezi secţiunea 2.3.2).

Index

C
literă 12
limbaj de programare 15, 22

LaTeX 30

```
\begin{theindex}
\item C
\subitem liter\u a 12
\subitem limbaj de programare 15, 22
\item \LaTeX\ 30
\end{theindex}
```

Pentru a construi în acest mod un index, ar trebui să căutăm cuvintele în toată cartea pe care o scriem, pentru a găsi numărul paginii. LaTeX oferă o serie de comenzi care ne scutesc de această muncă:

- `\index{`*text*`}` scrie comanda `\indexentry{`*text*`}{`*pagină*`}` în fişierul `idx`, unde *pagină* reprezintă numărul paginii unde apare comanda `\index`. Comanda nu poate să apară ca argument al altei comenzi dacă *text* conţine alte caractere în afară de litere, cifre şi semne de punctuaţie.

  Comanda `\index` nu are nici un efect dacă nu a fost dată comanda `\makeindex`.

- `\makeindex` autorizează scrierea tuturor intrărilor `\indexentry` create de comanda `\index` în fişierul `idx`. Această comandă trebuie să apară în preambulul documentului.

Fişierul `.idx` este destinat utilitarului `makeindex`, care citeşte comenzile `\indexentry` şi generează un fişier cu extensia `.ind` care conţine un cadru `theindex` şi comenzi `\item`, `\subitem` şi `\subsubitem` corespunzătoare comenzilor `\indexentry` ordonate alfabetic.

Textul *text* care se transmite ca parametru comenzii `index` poate avea următoarea structură:

- *text!text1* caz în care *text1* va fi considerat un `subitem` la *text.*
- *text!text1!text2* caz în care *text2* va fi considerat un `subsubitem` la *text.*
- *text@text1* această structură are următoarea semnificaţie: la ordonarea alfabetică se va ţine cont de *text* dar în fişierul `ind` se va scrie *text1.*
- *text*`|(` specifică faptul că text apare pe mai multe pagini şi se doreşte scrierea lor (în index) ca domeniu numeric în dreapta lui *text*, comanda trebuie să fie urmată de perechea ei *text*`|)`.
- caracterele !, @ şi | îşi pierd semnificaţia de mai sus dacă sunt precedate de ghilimele. Deci pentru a putea scrie `@` în index va trebui să-l transmitem comenzii `\index` ca `"@`.

Fişierul `.ind` rezultat în urma procesării cu `makeindex` a fişierului `.idx` poate fi inclus la o a doua procesare în sursă cu comanda `\printindex`, pentru care în lista de opţiuni ale comenzii `\documentstyle` trebuie să apară opţiunea `makeidx`.

Pentru exemplul de mai jos vom procesa mai întâi sursa cu LaTeX. Se va genera un fişier cu acelaşi nume ca al sursei, dar cu extensia `.idx`. Vom rula utilitarul `makeindex`, de exemplu pentru pachetul *emTEX*:

makeindx nume_fişier

după care mai procesăm sursa o dată.

Index

β, 5
α, 9
alpha, 8–21, 30
  litera, 10

```
\documentstyle[...,makeidx,...]{...}
\makeindex
...
Pag.  5: \index{$\beta$}
Pag.  8: \index{alpha|(}
Pag.  9: \index{alpha@$\alpha$}
Pag. 10: \index{alpha!litera}
Pag. 21: \index{alpha|)}
Pag. 30: \index{alpha}
...
\printindex
```

Folosind în lista de parametri opţionali ai comenzii `\documentstyle` opţiunea `showidx`, comanda `\index{`*cuvânt*`}` va plasa şi o notă marginală formată din *cuvânt* la pagina la care apare această comandă. Notele marginale de pe o pagină sunt puse una sub cealaltă, şi nu în dreptul locului în care apare comanda `\index`.

Uneori se doreşte şi construirea unui glosar de termeni. Cadrul folosit va fi tot `theindex`, dar se recomandă folosirea comenzilor:

- `\glossary{`*text*`}` scrie comanda `\glossaryentry{`*text*`}{`*pagină*`}` în fişierul `glo`, unde *pagină* reprezintă numărul paginii unde apare comanda `\glossary`. Comanda nu poate să apară ca argument al altei comenzi numai dacă *text* conţine doar litere, cifre şi semne de punctuaţie.

  Comanda `\glossary` nu are nici un efect dacă nu a fost dată comanda `\makeglossary`.

- `\makeglossary` autorizează scrierea intrărilor `\glossaryentry` create de comanda `\glossary` în fişierul `glo`. Comanda `\makeglossary` trebuie să apară în preambulul documentului.

Fişierul `.glo` astfel rezultat va putea fi folosit ca punct de plecare pentru glosar. De regulă acesta va fi procesat separat.

Aceste comenzi sunt utile când pentru aceeaşi lucrare se generează şi un index şi un glosar.

Capitolul 5

Macrodefiniţii

Multe compilatoare lucrează cu macrodefiniţii, numite pe scurt macrouri. În esenţă, un macrou este o secvenţă de caractere care în timpul procesării va fi înlocuită cu definiţia macroului. Majoritatea comenzilor discutate până acum sunt macrouri. LaTeX este de fapt un pachet de macrouri.

Definiţia unui macrou poate fi formată din text ordinar, alte macrouri şi/sau primitive TeX. Primitivele TeX sunt secvenţe de control recunoscute de TeX, care în general nu se pot înlocui cu alte comenzi. Într-o analogie cu un limbaj de programare, aceste primitive pot fi comparate cu instrucţiuni de nivel scăzut, în timp ce macrourile pot fi comparate cu instrucţiunile de nivel înalt, care se bazează desigur pe cele de nivel scăzut.

Comanda prin care definim un nou macrou este:

`\newcommand{\`*comandă_nouă*`}{`*definiţie*`}`

prin care asociem noii comenzi `\`*comandă_nouă* definiţia *definiţie*. De exemplu:

... şi scoase iar un hohot: Hahahahihihihohohoho. După un minut, se auzi iar: Hahahahihihihohohoho.

Acest Hahahahihihihohohoho era inconfundabil.

```
\newcommand{\haha}{Hahahahihihihohohoho}
... \c si scoase iar un hohot: \haha.
Dup\u a un minut, se auzi iar: \haha.

Acest \haha\ era inconfundabil.
```

Rezultă deci că macrourile sunt utile atunci când o secvenţă mai lungă apare adesea în lucrare, caz în care o asociem unei comenzi cu nume mai scurt. Definiţia unui macrou poate conţine şi alte macrouri. De exemplu, în lucrarea noastră apare destul de des semnul:

care poate fi definit astfel:

```
\newcommand{\accesint}{
  \begin{picture}(00.00,00.00)
    \thicklines
    \put(-26.00,4.00){\circle{14.00}}
    \put(-26.00,4.00){\circle{11.00}}
    \put(-29.00,3.00){\rule{6.00\unitlength}{2\unitlength}}
    \put(-26.00,-9.50){\line(0,1){6.50}}
    \put(-30.00,-9.50){\line(1,0){8.00}}
  \end{picture}
}
```

De remarcat că la fiecare comandă `\accesint`, întreg cadrul `picture` va fi reprocesat, ceea ce este o pierdere de timp. De aceea, este indicată folosirea comenzilor `\newsavebox`, `\sbox` şi `\usebox` în locul macrourilor lungi fără parametru.

`\newcommand` are un efect local, ceea ce înseamnă că putem folosi o comandă doar în interiorul grupului în care a fost definită, sau în grupurile din interiorul acestuia.

Un grup este delimitat de două acolade sau de o pereche `\begin`, `\end`.

O comandă va începe întotdeauna cu caracterul \[1]. Ceea ce urmează după acest caracter este numele comenzii. Numele unei comenzi poate fi alcătuit dintr-o succesiune de oricâte litere sau dintr-un singur caracter care nu este literă.

LaTeX ignoră toate spaţiile ce urmează după o comandă al cărei nume este format din litere. De aceea, dacă dorim să tipărim un spaţiu după o astfel de comandă, trebuie să-l impunem cu comanda \␣.

Să presupunem că am definit comenzile `\haha` şi `\1`. Atunci nu putem scrie, de exemplu: `\hahaul se auzi` deoarece nu există comada `\hahaul`. Vom scrie `\haha ul se auzi` pentru a lipi definiţia comenzii `\haha` de "ul". Putem însă scrie `\haha-ul` deoarece caracterul `-` nu este o literă şi deci comanda este clar delimitată. Putem scrie, de asemenea, şi `\12`, care are ca efect tipărirea definiţiei comenzii `\1`, lipită de caracterul 2, deoarece nu se pune problema existenţei unei comenzi cu numele `12`. În schimb, `\1 2`

[1] Există o excepţie: caracterul special ~ este o comandă

va avea ca efect scrierea definiţiei comenzii \1 separată printr-un spaţiu de caracterul 2.

LaTeX permite definirea de macrouri cu parametri, urmând ca la folosirea macroului să fie specificate argumentele actuale care vor înlocui parametrii din definiţie.

Să presupunem că lucrăm la o carte de matematică în care apar des secvenţe ca $x_1, x_2, \ldots, x_n$, dar cu diferite nume ale elementelor. Putem folosi comanda \newcommand cu parametru opţional pentru a defini secvenţa de control \sir:

şirul $x_1, x_2, \ldots, x_n$ şi şirul

$$\alpha_1, \alpha_2, \ldots, \alpha_n$$

sunt convergente în virtutea ...

```
\newcommand{\sir}[1]{#1_1,#1_2,\dots,#1_n}
şirul $\sir{x}$ şi şirul
$$\sir\alpha$$
sunt convergente în virtutea ...
```

Sintaxa generală a comenzii \newcommand este deci:

\newcommand{*comanda*}[*n*]{*definiţie*}

unde n, dacă [n] este prezent, este un număr întreg care semnifică numărul de parametri ai comenzii. În cadrul definiţiei se vor folosi aceşti parametri cu ajutorul caracterului # urmat de numărul parametrului la care ne referim. La folosirea comenzii, această construcţie va fi înlocuită cu parametrul actual. O comandă poate avea maximum 9 parametri, adică n poate lua valori între 1 şi 9.

La folosirea comenzilor cu parametri, se va considera ca parametru primul caracter diferit de spaţiu sau comandă întâlnit, sau tot textul cuprins între acolade, dacă primul caracter diferit de spaţiu este o acoladă.

Atenţie! Acoladele care delimitează un parametru actual nu formează un grup. De exemplu:

Hahahaha nu e ceea ce doresc!
Hohohoho este ce vreau de fapt.

```
\newcommand{\haha}[1]{H#1h#1h#1h#1}
{\haha{\bf a} nu e ceea ce doresc!}\par
{\haha{{\bf o}} este ce vreau de fapt.}
```

Acest lucru se datorează faptului că în prima linie, substituţia se face astfel:

\haha{\bf a} → H\bf ah\bf ah\bf ah\bf a

în timp ce la a doua linie:

\haha{{\bf a}} → H{\bf a}h{\bf a}h{\bf a}h{\bf a}

Pentru a generaliza comanda \sir, ar fi poate mai potrivită definirea lui cu doi parametri:

şirul $x'_1, x'_2, \ldots, x'_n$ şi şirul mai lung

$$\alpha_1, \alpha_2, \ldots, \alpha_{n+1}$$

sunt convergente în virtutea ...

```
\newcommand{\sir}[2]
{#1_1,#1_2,\dots,#1_{#2}}
şirul $\sir{xn$ şi şirul mai lung
$$\sir\alpha{n+1}$$
sunt convergente în virtutea ...
```

În timpul procesării comenzii **\newcommand**, nu se face verificarea definiţiei. Se face doar verificarea caracterelor { şi }, pentru a stabili textul definiţiei, care va fi memorat ca atare. Acest text va fi procesat doar la folosirea comenzii declarate, după ce s-au înlocuit parametrii formali cu cei actuali — dacă este cazul.

Există două comenzi deosebit de utile dacă se doreşte începerea unui grup într-o comandă, şi — eventual — terminarea lui în alta:

\bgroup şi **\egroup**

Aceste comenzi sunt echivalente cu caracterele { respectiv }, dar au avantajul că nu sunt "văzute" în timpul procesării comenzii **\newcommand**. De exemplu:

Vom scrie **mic şi îngroşat** şi iar normal.

```
\newcommand{\bsb}{\bgroup\small\bf}
\newcommand{\esb}{\egroup}
Vom scrie \bsb mic \c si \^\i ngro\c sat
\esb \c si iar normal.
```

Faptul că în timpul procesării comenzii **\newcommand** nu se fac verificări asupra definiţiei are o consecinţă uneori foarte utilă, şi anume că putem folosi în aceste definiţii comenzi care încă nici nu sunt definite. Important este ca la folosirea comenzii, acestea să fie definite:

... Acesta este un exemplu cam absurd!!!

```
\newcommand{\treixxx}{\xxx\xxx\xxx}
...\newcommand{\xxx}{!}
Acesta este un exemplu cam absurd\treixxx
```

Textul unei definiţii poate conţine bineînţeles şi comanda **\newcommand**. Putem astfel defini o comandă a cărei efect să fie definirea unei alte comenzi. Dacă dorim să definim un macrou cu doi parametri al cărei efect să fie definirea primului parametru ca al doilea repetat de trei ori, putem scrie:

Nu!Nu!Nu!

```
\newcommand{\tridef}[2]{
  \newcommand{#1}{#2#2#2}}
\tridef{\nuuu}{Nu!} \nuuu
```

Pentru lucrări mai mari, este întotdeauna utilă definirea unor macrouri pentru uşurarea scrierii. Chiar dacă este vorba doar de a prescurta o comandă mai lungă care apare des în lucrare. De exemplu, dacă folosim foarte des ecuaţii numerotate, în loc să scriem de fiecare dată

```
\begin{equation}\label{...}
....
\end{equation}
```

putem defini macroul `\ec`:

```
\newcommand{\ec}[2]
    {\begin{equation}\label{#1}#2\end{equation}}
```

cu ajutorul căruia scrierea unei ecuaţii se face mai simplu. Primul parametru al comenzii reprezintă eticheta prin care ne vom referi la ecuaţie, iar al doilea este formula ecuaţiei.

Dacă scriem foarte des lucrări care ar putea folosi în comun unele macrouri, putem pune toate aceste macrouri într-un fişier pe care îl putem include în fişierul la care lucrăm cu ajutorul comenzii:

`\input{`*nume_fişier*`}` sau

`\input` *nume_fişier*

unde *nume_fişier* este numele, inclusiv extensia, fişierului ce conţine macrourile. Fişierul sursă va fi procesat ca şi când în locul comenzii `\input` ar apărea textul din fişierul inclus.

O metodă mai elegantă este scrierea unei noi opţiuni de stil. Pentru un exemplu, vezi anexa B.

Numele comenzilor sunt identificatori unici, ceea ce înseamnă că nu pot exista mai multe comenzi cu acelaşi nume, chiar dacă au un număr diferit de parametri.

Din acest motiv, comanda `\newcommand` va da eroare dacă primul ei parametru este un macrou deja definit. Dacă se doreşte schimbarea definiţiei unui macrou, trebuie folosită comanda

`\renewcommand{\`*comandă_veche*`}[`*n*`]{`*definiţie nouă*`}`

unde `\`*comandă_veche* trebuie să fie un macrou deja definit. Şi comanda `\renewcommand` are un efect local, adică noua definiţie rămâne valabilă doar în grupul în care a fost dată comanda `\renewcommand`.

... Hohohoho Hahahaha ...

```
\newcommand{\haha}{Hahahaha}...
{\renewcommand{\haha}[1]{H#1h#1h#1h#1}
\haha{o}} \haha\ ...
```

Cu toate că nu este documentată în manualul de referinţă al LaTeX-ului [2], majoritatea sistemelor LaTeX pun la dispoziţie opţiunea de stil **ifthen** ce conţine definiţia a două macrouri deosebit de puternice: **\ifthenelse** şi **\whiledo**. Sintaxa generală a acestor comenzi:

\ifthenelse{*test*}{*ramura da*}{*ramura nu*}

\whiledo{*test*}{*corp*}

care implementează instrucţiunea condiţională şi instrucţiunea repetitivă specifice oricărui limbaj de programare de nivel înalt. Argumentul *test* poate fi una din următoarele construcţii:

- O relaţie între doi întregi, formată cu ajutorul operatorilor <, > sau =. Întregii pot fi constante sau valori ale unor contoare. Exemplu: **\value{page}>15**.
- Comanda **\equal**{*şir1*}{*şir2*}, care este adevărată dacă cele două şiruri de caractere sunt identice. Dacă *şir1* sau/şi *şir2* conţin macrouri, acestea sunt mai întâi expandate (adică înlocuite cu definiţia lor). Comparaţia va fi făcută între şirurile rezultate.
- O combinaţie logică a celor două construcţii de mai sus, folosind operatorii **\or**, **\and**, **\not** şi parantezele **\(** şi **\)**. Exemplu:

  ```
  \not \( \equal{\haha}{Hahahaha} \or \value{figure} >5 \)
  ```

Avem practic două comenzi în una: Prima variantă. A doua variantă.

```
\newcommand{\test}[1]{
\ifthenelse{#1=1}{Prima variantă}
{A doua variantă}}
Avem practic dou\u a comenzi \^\i n una:
\test1. \test2.
```

Pentru o exemplificare a comenzii **\whiledo**, vezi anexa C.

Unele comenzi LaTeX dau eroare dacă sunt folosite în argumentele anumitor comenzi. De exemplu, folosirea comenzii **\ref** în argumentul unei comenzi **\caption** produce eroare. Spunem despre comanda **\ref** că este fragilă. Comenzile fragile nu pot fi folosite în argumente mobile, cum este cazul argumentelor comenzilor **\caption**. Argumente mobile sunt cele care nu sunt plasate în pagină de îndată ce sunt procesate, cum este cazul argumentelor comenzilor **\footnote**, **\marginpar**, **\thanks**, comenzi secţionale.

Dacă se doreşte folosirea unei comenzi fragile într-un argument mobil, ea trebuie protejată cu comanda `\protect`. De exemplu, pentru figura **1.1** de la pagina 2 am dat următoarea comandă:

```
\caption{{\bf Un exemplu simplu de articol scris în \LaTeX.
Rezultatul procesării acestui fişier este dat în figura
\protect\ref{exout}.}}
```

`exout` fiind eticheta figurii **1.2**.

Comenzi fragile sunt: `\(`, `\)`, `\[`, `\]`, `\begin`, `\end`, `\footnote`, `\\`, `\item`, precum şi toate comenzile care au şi o formă cu `*`, şi cele care acceptă parametri opţionali.

•

La scrierea documentelor în LaTeX, se folosesc foarte des cadruri. Avem posibilitatea de a defini cadruri noi, atunci când cele predefinite nu pot rezolva problema, cu comenzile:

`\newenvironment{`*nume_cadru*`}[`*n*`]{`*def1*`}{`*def2*`}`

respectiv

`\renewenvironment{`*nume_cadru*`}[`*n*`]{`*def1*`}{`*def2*`}`

unde *nume_cadru* reprezintă numele cadrului pe care dorim să îl definim, respectiv redefinim, *n* reprezintă numărul de parametri ai cadrului (`[`*n*`]` poate lipsi), iar *def1* şi *def2* sunt două definiţii, prima fiind procesată la începutul unui cadru *nume_cadru*, cealaltă la sfârşitul lui. Doar în *def1* se pot folosi eventualii parametri.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

```
\newenvironment{matrice}[1]{$$#1=\left(
\begin{array}{ccc}}{\end{array}\right)$$}
...
\begin{matrice}{A}
a_{11} & a_{12} & a_{13}\\
a_{21} & a_{22} & a_{23}\\
a_{31} & a_{32} & a_{33}
\end{matrice}
```

Comanda `\newenvironment` va da eroare dacă *nume_cadru* este deja definit, iar `\renewenvironment` va da eroare dacă este nedefinit.

Sunt rare cazurile când avem nevoie de noi cadruri. LaTeX ne oferă însă o facilitate mult mai des folosită de cei care scriu texte ştiinţifice, şi anume comanda:

`\newtheorem{`*nume_cadru_teoremă*`}{`*nume teoremă*`}[`*contor*`]`

sau:

`\newtheorem{`*nume_cadru_teoremă*`}[`*nume_cdr_teor_echiv*`]{`*nume teoremă*`}`

unde:

nume_cadru_teoremă este numele cadrului în care vom scrie conţinutul teoremei.

nume teoremă este numele teoremei (Teorema, Propoziţia, Corolar etc.).

contor este un contor care va face parte din numărul teoremei, şi de fiecare dată când acest contor este modificat, se resetează numărul teoremei la zero. Astfel, dacă se va specifica aici contorul `section`, atunci în secţiunea 3 din capitolul 2, teoremele vor fi numerotate cu 2.3.1, 2.3.2 etc..

nume_cdr_teor_echiv este numele cadrului unei teoreme deja definite. Teorema pe care o definim va folosi pentru numerotare contorul celei echivalente.

Ca şi ecuaţiile, teoremele pot fi etichetate cu comanda `\label` pentru a se putea face referiri la ele:

Teorema 5.1 *Dacă în triunghiul ABC $\hat{B}$ este unghi drept, atunci $||AB||^2+||BC||^2 = ||CA||^2$.*

Propoziţia 5.2 *Teorema (5.1) este valabilă doar în spaţii euclidiene.*

```
\newtheorem{teorema}{Teorema}[chapter]
\newtheorem{prop}[teorema]{Propozi\c tia}
...
\begin{teorema}\label{pitagora}
Dac\u a \^\i n triunghiul $ABC$ $\hat B$
este unghi drept, atunci $||AB||^2+
||BC||^2=||CA||^2$. \end{teorema}
...
\begin{prop} Teorema (\ref{pitagora}) este
valabil\u a doar \^\i n spa\c tii
euclidiene. \end{prop}
```

Ca un exemplu general pentru comenzile prezentate în acest captiol, am construit un cadru nou care să ne permită o scriere comodă a teoremelor în următorul format:

Teorema 5.1 (Pitagora): *Dacă în $\triangle ABC$ unghiul $\widehat{ABC}$ este drept, atunci $||AB||^2 + ||BC||^2 = ||CA||^2$.*

Demonstraţie: Fie $a = ||AB||$, $b = ||BC||$, $c = ||AC||$ şi $\alpha = \widehat{BAC}$. Atunci $a = c\cos\alpha$ şi $b = c\sin\alpha \Longrightarrow a^2 + b^2 = c^2\left(\sin^2\alpha + \cos^2\alpha\right) \Longrightarrow c^2 = a^2 + b^2$ ■

Iată mai jos sursa care a produs teorema 5.1:

```
\newcounter{teonum}[chapter]\setcounter{teonum}{0}
\renewcommand{\theteonum}{\arabic{chapter}.\arabic{teonum}}
\newenvironment{teonum}[1]{
                          % prima definitie: inceputul cadrului.
  \par\bigskip\noindent   % paragraf nou fara aliniat.
  \refstepcounter{teonum} % incrementam contorul teonum, si eticheta
                          % cadrului (daca exista) va primi valoarea
                          % lui \theteonum
  \bgroup                 % incepem un grup.Comanda definita mai jos
                          % va fi locala acestui grup.
  \newcommand{\demo}{     % comanda \demo va delimita enuntul de
                          % demonstratie. Definitia ei este:
    \egroup\par\smallskip\noindent % sfarsitul grupului ce delimi-
                          % teaza enuntul,paragraf nou fara aliniat.
    {\it Demonstra\c tie: } % scriem cursiv "Demonstratie",
    \bgroup               % si incepem un grup ce va delimita
                          % demonstratia.
  }                       % sfarsitul definitiei comenzii \demo
  {\bf Teorema \theteonum\ (#1): } % Scriem ingrosat "Teorema",
                          % numarul ei si parametrul cadrului.
  \bgroup\it              % incepem grupul ce delimiteaza enuntul.
}{                        % sfarsitul primei definitii, inceputul
                          % celei de-a doua
  \rule{0.5em}{0.5em}     % un patratel negru ce semnifica sfarsitul
                          % demonstratiei.
  \egroup                 % sfarsitul grupului ce delimiteaza
                          % demonstratia
  \egroup\par\bigskip     % sfarsitul grupului ce delimiteaza cadrul
}                         % sfarsitul celei de-a doua definitii.

\begin{teonum}{Pitagora}\label{pitag1}
Dac\u a \^\i n $\triangle ABC$ unghiul $\widehat{ABC}$ este drept,
atunci $||AB||^2+||BC||^2=||CA||^2$.
\demo Fie $a=||AB||$, $b=||BC||$, $c=||AC||$ \c si
$\alpha=\widehat{BAC}$. Atunci $a=c\cos\alpha$ \c si $b=c\sin\alpha$
$\Longrightarrow$ $a^2+b^2=c^2\left(\sin^2\alpha+
\cos^2\alpha\right)$ $\Longrightarrow$ $c^2=a^2+b^2$
\end{teonum}

Iat\u a mai jos sursa care a produs teorema \ref{pitag1}:
```

Atenţie! În sintaxa comenzilor **\newcommand**, **\newenvironment** şi **\newtheorem** este interzisă folosirea spaţiilor între argumente. De exemplu: **\newcommand{\test}{Test}** şi nu **\newcommand{\test} {Test}**.

Capitolul 6

Modul paragraf

Atunci când TEX este în mod paragraf, el citeşte text din fişierul de intrare şi formează rânduri, cu care formează paragrafe. Acest lucru îl realizează considerând toate obiectele (în general litere) ca pe nişte „cutii“ pe care le alătură. Când rândul este plin, următoarea cutie este plasată pe următorul rând etc.. Deoarece de regulă aceste „cutii“ nu încap exact pe un rând, TEX întinde sau comprimă spaţiile dintre ele. În cazuri rare, el poate întinde exagerat de mult spaţiile, generând un avertisment „Underfull hbox ...“, iar alteori nu poate comprima suficient de mult spaţiile, astfel că textul nu va mai fi aliniat corespunzător în marginea din dreapta; în acest caz, se generează un mesaj de avertizare „Overfull hbox ...“ (vezi capitolul 10). În general aceste probleme pot fi rezolvate sau prin indicarea unei despărţiri mai bune în silabe cu ajutorul comenzii `\-`, sau prin reformularea textului.

Elementele de bază cu care lucrează TEX în acest mod sunt caracterele, cel mai adesea literele.

6.1 Tipuri şi dimensiuni de caractere

În LATEX sunt disponibile mai multe tipuri de caractere (fonturi), reprezentate în ceea ce urmează prin comanda care duce la selectarea lor:

- `\rm` selectează caracterele din setul *roman* (set implicit pentru LATEX);
- `\it` selectează caracterele din setul *italic*;
- `\bf` selectează caracterele din setul *boldface* (caractere îngroşate);
- `\sf` setul de caractere *sans serif*;

- `\sc` setul de caractere *small capitals* are ca efect scrierea fiecărei litere ca literă mare, dar majusculele au dimensiuni mai mari.
- `\sl` selectează caracterele din setul *slanted* (caracterele sunt înclinate);
- `\tt` caracterele tip maşină de scris (*typewriter*);
- `\cal` caractere caligrafice (*caligraphic*), dar numai literele mari şi numai în mod matematic (vezi paragraful 7.3);

Acesta este stilul roman
Acesta este stilul italic
Acesta este stilul bold face
Acesta este stilul sans serif
ACESTA ESTE STILUL SMALL CAPS
Acesta este stilul slanted
Acesta este stilul typewriter
Se poate *evidenţia* textul

```
\rm Acesta este stilul roman\\
\it Acesta este stilul italic\\
\bf Acesta este stilul bold face\\
\sf Acesta este stilul sans serif\\
\sc Acesta este stilul small caps\\
\sl Acesta este stilul slanted\\
\tt Acesta este stilul typewriter\\
Se poate {\em eviden\c tia} textul
```

O parte din aceste fonturi este preîncărcată, iar altele sunt încărcate numai la cerere, atunci când în textul sursă apare comanda de trecere la respectivul font (vezi tabelul **6.1**). Un anumit font este activ atâta timp cât nu a apărut o altă comandă de modificare a tipului sau, în cazul folosirii delimitatorilor, atâta timp cât suntem în zona delimitată de aceştia. Delimitarea se face în modul următor: `{\tip text}`.

Pe lângă comenzile de schimbare a fonturilor mai există o comandă care nu defineşte un nou tip de font ci doar alege între fonturile `\it` şi `\rm`, aceasta este comanda `\em`. Acţiunea sa se rezumă doar la schimbarea fontului curent din roman în italic şi invers. Comanda se foloseşte la scoaterea în evidenţă a unui anumit text (vezi ultimul rând din exemplul de mai sus), acţiunea ei fiind restrânsă cu ajutorul delimitatorilor { şi }.

Declararea de noi fonturi se face cu comanda `\newfont` cu următoarea sintaxă:

`\newfont{\`*numefont*`}{`*font1*`}`

şi are ca efect declararea lui `\`*numefont* ca fiind comanda de selectare a fontului `font1`. În afară de aceasta se poate accesa şi numai un singur caracter dintr-un font cu comanda `\symbol{cod}` unde `cod` este codul ASCII al caracterului în zecimal (`'cod` este codul ASCII în octal iar,`"cod` este codul ASCII în hexazecimal).

	\it	\bf	\sl	\sf	\sc	\tt
5pt	N	C	N	N	N	N
6pt	N	C	N	N	N	N
7pt	P	C	N	N	N	N
8pt	P	C	C	C	C	C
9pt	P	P	C	C	C	P
10pt	P	P	P	P	C	P
11pt	P	P	P	P	C	P
12pt	P	P	P	P	C	P
14pt	C	P	C	C	C	C
17pt	C	P	C	C	C	C
20pt	C	C	C	C	C	C
25pt	N	C	N	N	N	N

Tabelul 6.1: Clase de fonturi: P = preîncărcate, C = încărcate la cerere, N = nu sunt disponibile.

Caracterele din aceste fonturi sunt disponibile (pentru o anumită opţiune dimensională a comenzii `\documentstyle`) în 10 dimensiuni diferite[1] (vezi tabelul **6.2**). Alegerea unei dimensiuni sau alta făcându-se cu comenzile corespunzătoare (`\tiny`, `\scriptsize`,`\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`). Toate aceste comenzi selectează automat fontul roman.

6.2 Simboluri speciale

Datorită faptului că anumite caractere din setul ASCII standard au semnificaţie în LaTeX, ele nu pot fi obţinute prin simpla scriere a lor în textul sursă, ci prin inserarea unor comenzi care au ca rezultat caracterul dorit. Aceste comenzi nu sunt disponibile în stilul `\tt`.

Caracterele speciale cu semnificaţia lor LaTeX sunt:

- # se foloseşte la definirea de comenzi LaTeX, şi anume `#n` se referă la parametrul numărul `n` al unei comenzi. Ca şi caracter se poate obţine cu ajutorul comenzii `\#`;

[1] Dimensiunile sunt date în puncte (72,27pt=1in, 1in=25,4mm).

dimensiunea	implicit (10pt)	opţiunea 11pt	opţiunea 12pt
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

Tabelul 6.2: Dimensiunea fonturilor.

- $ semnifică începutul sau sârşitul modulului matematic *"în text"*. Caracterul este produs de comanda **\$**;
- % semnifică începerea unui comentariu pe rândul pe care se găseşte; are ca efect ignorarea caracterelor care apar după el pe acel rând. Se obţine, ca şi caracter, cu comanda **\%**
- & se foloseşte pentru a separa coloanele într-un cadru `tabular` sau `array`, poate fi obţinut ca şi caracter cu comanda **\&**;
- _ (underscore) caracterul *underscore* (liniuţa de subliniere) se foloseşte în mod matematic pentru scrierea indicilor inferiori. Se poate obţine cu comanda **_**;
- { , } sunt folosite pentru delimitarea zonei de acţiune a unor comenzi sau pentru transmiterea de parametri comenzilor. Se obţin ca şi caractere cu comenzile **\{** respectiv **\}**;
- \ semnifică întotdeauna începutul unei comenzi şi se poate obţine ca şi caracter cu comanda `\backslash`, procesată în mod matematic sau cu comanda `\tt \char 39`;
- <,> sunt tipărite în mod paragraf ca şi ¡, respectiv ¿. Cele două caractere se pot obţine în mod matematic astfel: **$>$** pentru caracterul > respectiv **$<$** pentru <.

Pe lângă aceste caractere speciale mai există o serie de comenzi simple care permit accentuări ale altor caractere pentru a obţine litere specifice anumitor limbi (vezi tabelul **6.3**).

\‘{o}	ò	\~{o}	õ	\v{o}	ǒ	\d{o}	o̩
\’{o}	ó	\={o}	ō	\H{o}	ő	\b{o}	o̲
\^{o}	ô	\.{o}	ȯ	\t{oo}	o͡o	\i	ı
\"{o}	ö	\u{o}	ŏ	\c{o}	o̧		

Tabelul 6.3: Accente în mod paragraf (aplicate în exemplu literelor o sau i)

Există o serie de simboluri speciale care nu se obţin din caracterele obişnuite prin accentuarea lor. În tabelul **6.4** sunt prezentate caracterele disponibile indiferent de mod, iar în tabelul **6.5** sunt prezentate cele specifice modului paragraf. Acestea din urmă pot fi incluse în mod matematic doar ca parametri ai comenzii `\mbox`.

\dag	†	\S	§	\copyright	©
\ddag	‡	\P	¶	\pounds	£

Tabelul 6.4: Simboluri speciale (orice mod)

\aa	å	\AA	Å	\ae	æ	\AE	Æ
\o	ø	\O	Ø	\oe	œ	\OE	Œ
\l	ł	\L	Ł	>	¿	<	¡
\ss	ß						

Tabelul 6.5: Simboluri speciale (numai în mod paragraf)

6.3 Gestionarea rândurilor şi a paginilor

6.3.1 Gestionarea rândurilor

În cadrul unui paragraf textul este dispus pe rânduri în aşa fel încât toate rândurile să fie pline (cu excepţia ultimului, care, după caz, poate să fie doar parţial ocupat de text), iar distanţa dintre cuvinte să aibă o anumită valoare. Pentru a se putea realiza aceasta, LaTeX se va folosi şi de regulile de despărţire în silabe. În cazul în care la sfârşitul unui rând există un cuvânt prea lung pe care nu ştie să-l despartă în silabe, iar rândul are lungimea mai mare decât lungimea maximă admisă (specificată prin comanda `\textwidth`), va apărea mesajul de eroare *Overfull* **xxxx***pt to wide in line ...* (vezi capitolul 10). Remedierea se poate face în două moduri, şi anume: fie punând la începutul documentului comanda `\sloppy` care va face ca LaTeX să nu fie atât de rigid cu spaţierea cuvintelor (comanda cu efect invers este `\fussy`), fie spunându-i cum să despartă în silabe acel cuvânt (vezi secţiunea 2.3.3).

Perechea de comenzi `\sloppy` - `\fussy` funcţionează similar cu cadrul `sloppypar`.

Trecerea de la un paragraf la altul se face lăsând în textul sursă un rând liber.

Trecerea la o linie nouă se face cu comenzi specifice care obligă LaTeX să facă acest lucru. Aceste comenzi sunt:

- `\\` cu formele sale `\\[lungime]` şi `\\*[lungime]` are ca efect trecerea la un nou rând respectiv trece la un nou rând şi lasă un spaţiu egal cu `lungime` între cele două rânduri. Forma cu * inhibă trecerea la pagină nouă înainte de noul rând;
- `\newline` echivalentă cu `\\`;
- `\linebreak` forţează TeX să treacă la linie nouă;
- `\linebreak[n]`;n=0÷4 cere trecerea la linia nouă acţionând în felul următor: `\linebreak[0]` nici nu încurajează nici nu descurajează trecerea la o nouă linie, pe când comanda `\linebreak[4]` obligă trecerea la o linie nouă exact ca `\linebreak`, celelalte numere (1,2,3) reprezentând grade intermediare între acestea două;
- `~` (unbreakable space) interzice ruperea unei linii în acel loc;
- `\nolinebreak` interzice ruperea liniei în punctul în care apare comanda;

- `\nolinebreak[n]`; n=0÷4 transformă interdicţia comenzii de mai sus într-o sugestie astfel: cu cât n este mai mare cu atât sugestia este mai puternică. Comanda `\nolinebreak[0]` este echivalentă cu comanda `\linebreak[0]`, iar comanda `\nolinebreak[4]` este echivalentă cu comanda `\nolinebreak`;

Comenzile `\\` şi `\newline` nu au ca efect generarea mesajului `Underfull \hbox xxxpt in line ...`, deoarece trecerea la linie nouă se face chiar unde apare comanda iar restul liniei rămâne goală. În cazul comenzii `\linebreak` acest mesaj apare deoarece după trecerea la linie nouă textul rămas pe acea linie este extins (prin lărgirea spaţiilor dintre cuvinte) la întreaga linie.

Toate aceste comenzi pot fi folosite doar în mod paragraf. Dacă după aceste comenzi următorul caracter este `[`, el trebuie pus între acolade pentru a nu fi interpretat ca făcând parte din forma `[n]` a comenzii.

6.3.2 Gestionarea paginilor

Ca şi în cazul trecerii la linie nouă s-ar putea ca TEX să nu găsească locul cel mai bun de a trece la pagină nouă, ceea ce va avea ca efect o mai mică ocupare cu text a paginii.

Pentru ca toate paginile să aibă aceeaşi lungime (necesară în cazul scrierii unei cărţi) se va folosi comanda `\flushbottom` a cărei efect este adăugarea de spaţiu vertical suplimentar astfel încât toate paginile să aibă aceeaşi lungime. Anularea efectului acestei comenzi se face cu ajutorul comenzii `\ruggedbottom`. În primul caz TEX va anunţa că a adăugat spaţiu suplimentar prin mesajul `Underfull \vbox ...` .

Comenzile de trecere la pagină nouă sunt analoage comenzilor de trecere la linie nouă (motiv pentru care s-a păstrat comanda `\newline`). Aceste comenzi sunt:

- `\newpage` analoagă cu `\newline`;

- `\pagebreak`,`\pagebreak[`n`]`; n=0÷4 sintaxa şi semnificaţia parametrului n este analoagă celei de la comenzile `\linebreak` respectiv `\linebreak[`n`]`;

- `\nopagebreak`,`\nopagebreak[`n`]`; n=0÷4 sintaxa şi semnificaţia parametrului n este analoagă celei de la comenzile `\nolinebreak` respectiv `\nolinebreak[`n`]`;

Există cazuri când trecerea la o nouă pagină într-un anumit punct nu poate fi oprită nici cu comanda **\nopagebreak**. În acest caz se va folosi comanda **\samepage** astfel:

\samepage{*linii_text*} unde *linii_text* sunt liniile de text care conţin punctul în care nu se doreşte trecerea la pagină nouă. După fiecare linie goală se va pune comanda **\nopagebrek**

O comandă similară cu **\newpage** este comanda **\clearpage** doar că fiecare figură şi tabel care au rămas pe dinafară sunt puse pe una sau mai multe pagini, separate, fără text.

Două comenzi **\newpage** sau **\clearpage** pe un rând sunt echivalente cu una singură. În cazul formatului pe două coloane comenzile **\newpage** şi **\pagebreak** încep o nouă coloană pe când comenzile **\clearpage** şi **\cleardoublepage** încep o nouă pagină.

6.4 Spaţii şi spaţieri

Multe spaţii (verticale şi orizontale) necesare la scrierea unui text sunt predefinite şi reprezintă parametrii de stil. De exemplu spaţiul dintre rânduri, dintre ecuaţii şi text etc. Pe lângă aceste spaţii mai există şi o serie de comenzi care permit adăugarea de spaţiu, vertical şi orizontal, suplimentar.

În ceea ce urmează vom explica modul de acţiune al comenzilor de spaţiere orizontală şi verticală.

- **-** linie de despărţire în silabe;
- **--** linie mai lungă folosită la domenii numerice (–);
- **---** linie folosită la punctuaţie (—);
- **\,** spaţiu mic folosit între ghilimelele simple şi duble;
- **\␣**[2] spaţiu intracuvânt;
- **\/** spaţiu folosit pentru a nu permite ligatura anumitor litere;
- **\@** spaţiu de sfârşit de propoziţie. Se foloseşte numai dacă următoarea propoziţie începe cu un caracter care nu este literă mare;
- **\frenchspacing** suprimă spaţiul suplimentar de după punctuaţie, chiar şi atunci când se foloseşte comanda **\@**;

[2] Comanda **\␣** trebuie înţeleasă ca **** urmat de caracterul spaţiu (␣)

- `\nofrenchspacing` anulează efectul comenzii `\frenchspacing`;
- `\noindent` suprimă alineatul la început de paragraf;
- `\par` este echivalentă cu o linie liberă în text;
- `\hspace{`*lun*`}` produce un spaţiu orizontal de lungime *lun*. Uneori, la început sau sfârşit de linie, acest spaţiu nu este luat în considerare;
- `\hspace*{`*lun*`}` ca la `\hspace` doar că, de data aceasta, comanda este luată în considerare indiferent de locul în care apare în paragraf;
- `\vspace{`*lun*`}` analog cu `\hspace` doar că spaţiul lăsat este vertical;
- `\vspace*{`*lun*`}` forma *robustă* a comenzii `\vspace` (analoagă cu `\hspace*`);
- `\hfill` spaţiu orizontal de lungime variabilă (atât cât lipseşte până la umplerea unui rând). Echivalent cu `\hspace{\fill}`;
- `\vfill` spaţiu vertical de lungime variabilă (atât cât trebuie până la umplerea pagini). Este echivalent cu `\vspace{\fill}`.
- `\dotfill` are acelaşi mod de acţiune ca şi comanda `\fill` doar că produce puncte;
- `\hrulefill` acţionează la fel ca `\dotfill` dar produce o linie orizontală;
- `\smallskip`,`\medskip`,`\bigskip` spaţii verticale ce depind de tipul documentului echivalente cu comenzile `\hspace{\smallskipamount}`, `\hspace{\medskipamount}` şi `\hspace{\bigskipamount}`, aceste comenzi sunt predefinite şi valoarea lor depinde de tipul documentului;
- `\addvspace{`*lun*`}` adaugă un spaţiu vertical de lungime *lun*. Totuşi dacă această comandă s-a mai folosit pentru a adăuga spaţiu *lun1* din acelaşi punct ca şi comanda curentă, atunci comanda curentă va avea ca efect adăugarea de spaţiu *lun-lun1*, deci exact cât mai trebuie pentru ca spaţiul total adăugat de cele două comenzi să fie *lun*;

Oricare din comenzile de lungime prezentate pot primi ca parametri lungimi date în oricare din unităţile de lungime acceptate (vezi la pagina 90). Deci se acceptă atât mărimi date în `cm` cât şi lungimi date ca multipli şi submultipli ale unor comenzi de lungime. De exemplu `\vspace{5\parskip}` semnifică un spaţiu vertical de 5 ori `parskip`.

Comanda *hspace* lasă spaţii orizontale.
Numele_____Prenumele _____

Nr.....din .....

Acest paragraf începe fără alineat.

```
Comanda {\em hspace} las\u a
\hspace{2cm} spa\c tii
orizontale.\\
Numele\hrulefill
Prenumele \hrulefill\\[1cm]
Nr\dotfill din \dotfill\hfill
\vspace{1cm}

\noindent Acest paragraf începe
f\u ar\u a alineat.
```

6.5 Note marginale

Pentru a putea face comentarii pe marginea unei pagini avem la dispoziţie comanda `\marginpar` cu următoarea sintaxă:
`\marginpar[`*text_stânga*`]{`*text_dreapta*`}` sau
`\marginpar{`*text_dreapta*`}`

Această comandă va scrie textul *text_stânga* ca notă marginală dacă comanda apare pe pagina din stânga şi repectiv textul *text_dreapta* dacă apare pe o pagină din dreapta. A doua formă are ca efect plasarea lui *text_dreapta* ca notă marginală indiferent de pagina pe care apare. Indiferent de forma folosită şi de stilul documentului nota marginală se plasează întotdeauna pe marginea liberă a foii de hârtie (cea care după legarea lucrării ar rămâne liberă).

În cazul tipăririi pe o singură faţă nota marginală este plasată întotdeauna pe marginea din dreapta a colii scrise, iar în cazul tipăririi pe două feţe (opţiunea `twoside`) nota este plasată pe marginea din dreapta pentru paginile impare şi pe cea din stânga pentru paginile pare. În cazul tipăririi pe două coloane (opţiunea `twocolumn`) nota este plasată pe marginea foii de hârtie cea mai apropiată coloanei în care apare comanda.

Exceptând cazul tipăririi pe două coloane, dispunerea notelor marginale poate fi modificată cu comenzile:

- `\reversemarginpar` care are ca efect plasarea notei în marginea opusă celei implicite;
- `\normalmarginpar` revine la plasarea implicită anulând efectul comenzii `reversmarginpar` precedente;

Notă! Exemplu: undeva în acest alineat am introdus următoarea comandă

`\marginpar{Notă!}`.

Comanda `\footnote` se foloseşte pentru note de subsol. Acceptă un argument care va fi procesat în mod paragraf şi este plasat la baza paginii curente.

6.6 Dreptunghiuri pline

Se pot crea dreptunghiuri pline (umplute cu negru) cu ajutorul comenzii `\rule` cu sintaxa:

`\rule{`*dim_x*`}{`*dim_y*`}` sau
`\rule[`*dy*`]{`*dim_x*`}{`*dim_y*`}`

care creează un dreptunghi plin de lăţime *dim_x* şi înălţime *dim_y*. Dreptunghiul fiind înălţat faţă de baza textului cu *dy* (dacă acest parametru este prezent). În cazul în care lăţimea dreptunghiului *dim_x* este zero acesta nu mai este desenat dar i se rezervă spaţiul vertical ocupat de înălţimea sa, astfel TEX va ajusta spaţiul vertical pentru a lăsa loc pentru dreptunghi.

Exemplu:

Folosirea lui `\rule` ajută la ....
linie verticală de 2x10pt ▌, linie orizontală de 10x2pt ▬, linie de 4x18pt coborâtă cu 3pt sub nivelul rândului ▌.

```
Folosirea lui \verb|\rule| ajută la ....

linie verticală de 2x10pt
\rule{2pt}{10pt}, linie orizontală de
10x2pt \rule{10pt}{2pt}, linie de 4x18pt
cobor\^ată cu 3pt sub nivelul
r\^andului \rule[-3pt]{4pt}{18pt}.
```

Capitolul 7

Modul matematic

Formulele matematice pot să apară în două ipostaze: în cadrul textului (pe acelaşi rând împreună cu "textul ordinar") sau pe toată lăţimea paginii. Pentru a putea procesa astfel de formule trebuie să specificăm, prin comenzi speciale, felul în care dorim să apară o anumită formulă matematică. În cadrul formulelor matematice "în text" specificarea se face prin trei comenzi echivalente:

`$` *formula* `$`;

`\begin{math}` *formula* `\end{math}`;

`\(` *formula* `\)`;

Pentru a scrie formule în mod paragraf există următoarele comenzi echivalente:

`$$` *formula* `$$`;

`\begin{displaymath}` *formula* `\end{displaymath}`;

`\[` *formula* `\]`;

Aceste trei comenzi produc formule pe rânduri separate, dar nu le numerotează. Pentru a produce formule numerotate, vezi capitolul 4.

În mod matematic, toate caracterele din text sunt considerate simboluri matematice. Spaţiile sunt ignorate în acest mod, dar ele rămân totuşi necesare pentru delimitatea comenzilor.

7.1 Comenzi în mod matematic

În continuare vom prezenta comenzile mai frecvent folosite la scrierea (culegerea) formulelor matematice:

- _,^ specifică scrierea parametrului lor ca indice inferior respectiv superior;
- ' produce simbolul $'$ (prim);
- `\frac{`*numărător*`}{`*numitor*`}` reprezintă o fracţie, semnificaţia parametrilor fiind dată de însăşi numele lor;
- `\sqrt{`x`}`,`\sqrt[`n`]{`x`}` reprezintă rădăcina pătrată respectiv de ordinul n din x;
- există definite o serie de funcţii matematice care se obţin cu comenzile prezentate în tabelul **7.1** şi a căror semnificaţie este sugerată de numele comenzii;
- există două funcţii pentru operaţii modulo (`\bmod` şi `\pmod`) a căror acţiune este exemplificată mai jos;
- `\sum` generează simbolul $\sum$ (vezi exemplul de mai jos);
- `\int` generează simbolul $\int$ (vezi exemplul de mai jos);
- `\lim` simbolul folosit pentru scrierea limitei unui şir sau funcţii (vezi exemplul de mai jos);
- `\ldots` trei puncte în partea de jos a rândului (comanda este disponibilă în toate modurile);
- `\cdots`, `\vdots`, `\ddots` trei puncte la mijlocul rândului, verticale respectiv diagonale (sunt necesare în cazul scrierii de tablouri n-dimensionale);
- `\cal` este o declaraţie de tip care duce la scrierea celor 26 de litere, mari în stil caligrafic ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}, \ldots, \mathcal{Z}$);

$\sum_{i=0}^{n} A_i^k = \sqrt[3]{\frac{k+\sqrt{k+1}}{3}}$. Se ştie că

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n = e$$

Formula Leibnitz-Newton este:

$$\int_a^b f(x)dx = F(b) - F(a)$$

modulo: $n = m \bmod 256$, $h \equiv z \pmod{c+d}$.

```
$\sum_{i=0}^{n}A_{i}^{k}=\sqrt[3]{
    \frac{k+\sqrt{k+1}}{3}}$.
Se ştie că \[\lim_{n\rightarrow\infty}
\left(1+\frac{1}{n}\right)^{n}=e\]
Formula Leibnitz-Newton este:
\[\int_{a}^{b}f(x)dx=F(b)-F(a)\]
modulo: \(n=m \bmod 256\), \(h\equiv z
\pmod{c+d}\).
```

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

Tabelul 7.1: Funcţiile matematice

(	(	[	[	{	\{
[	\lbrack	⌊	\lfloor	⌈	\lceil
{	\lbrace	⟨	\langle		
)	)	]	]	}	\}
]	\rbrack	⌋	\rfloor	⌉	\rceil
}	\rbrace	⟩	\rangle		

Tabelul 7.2: Delimitatori

În tabelul **7.2** sunt prezentaţi delimitatorii disponibili în LaTeX cât şi comenzile care-i generează. Pentru ca un delimitator care apare într-o formulă să aibă dimensiunea corectă trebuie ca ei să fie precedaţi de comenzile `\left` respectiv `\right` după cum delimitatorul este de deschidere sau de închidere. Obligatoriu comenzile `\left` şi `\right` trebuie să apară în perechi. Totuşi există cazuri când delimitatorul nu are şi perechea sa, în acest caz comanda pereche `\left` sau `\right` va fi urmată de un punct (.).

Exemplu:

$$A = \left(\begin{array}{c} a_1 \\ \vdots \\ a_n \end{array} \right)$$

```
\[A=\left(\begin{array}{c}
              a_{1}\\
              \vdots\\
             a_{n}
          \end{array}\right) \]
```

$$f(x) = \left\{ \begin{array}{rl} x & \text{dacă } x > 0 \\ -x & \text{dacă } x \leq 0 \end{array} \right.$$

```
\[ f(x)=\left\{\begin{array}{rcl}
                x& \mbox{dac\u a}& x>0 \\
               -x& \mbox{dac\u a}& x\leq0
               \end{array}\right. \]
```

Comanda **\mbox{***text***}** are ca efect procesarea textului *text* în mod LR, deci permite scrierea textului în mod matematic.

Matricele se pot scrie în mod matematic folosind cadrul **array**. Sintaxa generală a acestei comenzi este:
\begin{array}[*locv*]{*mască*} *rânduri* **\end{array}**
unde *locv* reprezintă modul de dispunere al matricei pe verticală faţă de punctul curent. Valori posibile sunt **t (top)**, **b (bottom)** şi **c (center)** (implicit matricea va fi dispusă centrat). Parametrul *mască* reprezintă o matriţă pe coloane a tabloului reprezentată prin atâtea litere câte coloane există, unde literele pot fi oricare dintre *l (aliniere la stânga)*, *r (aliniere la dreapta)* şi *c (centrat)*. Matricea este formată din mai multe *rânduri* despărţite prin comanda ****. Fiecare rînd este format din mai multe coloane despărţite între ele prin caracterul **&**. Comanda **\end{array}** specifică sfârşitul cadrului.

Exemplu (vezi şi exemplul precedent):

$$\begin{array}{lc} x+y & z+c \\ & \boxed{\begin{array}{c} 37 \\ 5x \end{array}} \\ t & \end{array}$$

```
\[ \begin{array}{lc}
   x+y&z+c\\
   t&\framebox{\begin{array}[b]{c}
   37\\5x\end{array}}
   \end{array}\]
```

În mod matematic există posibilitatea de a pune deasupra şi dedesubtul unor expresii linii respectiv acolade cu ajutorul comenzilor:

- **\overline{***text***}** pune o linie peste textul *text*;
- **\underline{***text***}** subliniază textul *text*;
- **\overbrace{***obiect***}^{***obiect1***}** pune o acoladă peste obiectul *obiect* iar în centrul acoladei, centrat, pune obiectul *obiect1*;
- **\underbrace{***obiect***}_{***obiect1***}** ca şi la **\overbrace**, doar că de data aceasta totul se pune dedesubtul obiectului *obiect*;

- `\overbrace{`*obiect*`}`, `\underbrace{`*obiect*`}` pune doar acoladele deasupra respectiv dedesubtul lui *obiect*;

Este posibilă chiar şi „stivuirea" expresiilor matematice cu ajutorul comenzii `\stackrel{`*a*`}{`*b*`}`, având ca efect punerea lui *a* peste *b*, dimensiunea lui *a* fiind cea a unui indice superior.

Exemplu:

$$\overbrace{\overline{x+x+\ldots+x}}^{n \text{ ori}} = n\overline{x}$$

$$\vec{x} \stackrel{\text{def}}{\equiv} (x_1, x_2, \ldots, x_n)$$

```
\[ \underbrace{\overline{
x+x+\ldots+x}}_{n \mbox{ori}}=
n\overline{x}\]
\[\vec{x}\stackrel{\mbox{def}}
{\equiv}
(x_{1},x_{2},\ldots,x_{n})\]
```

În afara linilor şi acoladelor, deasupra expresiilor se mai poate pune şi simbolul de vector cu comanda `\vec{`*expresie*`}`. Această comandă a dus şi la următoarea problemă: dacă $\vec{i}$, $\vec{j}$ sunt doi versori atunci scrierea lor cu ajutorul comenzii `\vec` ar fi $\vec{i}$, $\vec{j}$, dar aceasta, în cel mai bun caz, duce la suprascrierea punctului (de la i respectiv j) peste săgeată, rezultatele obţinute fiind uneori nesatisfăcătoare. Pentru aceasta au fost create simbolurile i respectiv j fără punct cu ajutorul comenzilor `\imath` respectiv `\jmath`.

Accentele în mod matematic se obţin cu ajutorul comenzilor prezentate în tabelul **7.3**. Pentru comenzile `\hat{`*arg*`}` şi `\tilde{`*arg*`}` există şi versiunile mai largi `\widehat{`*arg*`}` şi `\widetilde{`*arg*`}` care încearcă să găsească simbolul corespunzător pentru a acoperi argumentele *arg*, totuşi nu pot fi oricât de largi.

Exemplu:

Fie $\hat{e}_1$ versorul axei x ...
În triunghiul ABC unghiul $\widehat{ABC}$ ...

```
Fie $\hat{e_{1}}$ versorul
axei $x$ ...

\^In triunghiul ABC unghiul
$\widehat{ABC}$ ...
```

Tabelele **7.4** şi **7.9** conţin diferite simboluri ce pot fi folosite în mod matematic.

Semnul $\int$ este un operator mare. El se comportă diferit faţă de un simbol oarecare în două privinţe: este înconjurat de un spaţiu mai mare decât simbolurile obişnuite, şi se poate opta pentru poziţionarea limitelor puse cu _ şi ^ deasupra respectiv sub operator, cu comanda `\limits`:

$$\int_a^b xdx = \frac{a^2 - b^2}{2}$$

```
$$\int_a^bxdx=\frac{a^2-b^2}{2}$$
```

$$\int\limits_a^b xdx = \frac{a^2 - b^2}{2}$$

```
$$\int\limits_a^bxdx=\frac{a^2-b^2}{2}$$
```

În tabelul **7.5** sunt prezentaţi toţi operatorii mari disponibili în LaTeX în mod matematic.

$\hat{a}$	\hat a	$\check{a}$	\check a	$\tilde{a}$	\tilde a	$\acute{a}$	\acute a
$\bar{a}$	\bar a	$\grave{a}$	\grave a	$\dot{a}$	\dot a	$\ddot{a}$	\ddot a
$\breve{a}$	\breve a	$\vec{a}$	\vec a				

Tabelul 7.3: Accente în mod matematic

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall	$\mho$	\mho
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists	$\Box$	\Box
$\imath$	\imath	∇	\nabla	$\neg$	\neg	$\Diamond$	\Diamond
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat	∞	\infty
ℓ	\ell	$\top$	\top	$\natural$	\natural	$\backslash$	\backslash
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp	$\spadesuit$	\spadesuit
$\Re$	\Re	$\|$	|	$\clubsuit$	\clubsuit	$\triangle$	\triangle
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit	∂	\partial
$\heartsuit$	\heartsuit						

Tabelul 7.4: Alte simboluri în mod matematic

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot	$\prod$	\prod
$\bigcup$	\bigcup	$\bigotimes$	\bigotimes	$\coprod$	\coprod	$\bigsqcup$	\bigsqcup
$\bigoplus$	\bigoplus	$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge				

Tabelul 7.5: Operatori mari

+	+	−	-	±	\pm	∩	\cap
∨	\vee	∓	\mp	∪	\cup	∧	\wedge
\	\setminus	⊎	\uplus	⊕	\oplus	·	\cdot
⊓	\sqcap	⊖	\ominus	×	\times	⊔	\sqcup
⊗	\otimes	*	\ast	⊘	\oslash	◁	\triangleleft
⋆	\star	▷	\triangleright	⊙	\odot	◇	\diamond
≀	\wr	†	\dagger	∘	\circ	◯	\bigcirc
‡	\ddagger	•	\bullet	∐	\amalg	△	\bigtriangleup
÷	\div	▽	\bigtriangledown				

Tabelul 7.6: Operatori binari

<	<	>	>	=	=	≤	\leq
≥	\geq	≡	\equiv	≺	\prec	≻	\succ
∼	\sim	⪯	\preceq	⪰	\succeq	≃	\simeq
≪	\ll	≫	\gg	≍	\asymp	⊂	\subset
⊃	\supset	≈	\approx	⊆	\subseteq	⊇	\supseteq
≅	\cong	⊑	\sqsubseteq	⊒	\sqsupseteq	⋈	\bowtie
∈	\in	∋	\ni	⋈	\Join	⊢	\vdash
⊣	\dashv	⊨	\models	⌣	\smile	∣	\mid
≒	\doteq	⌢	\frown	∥	\parallel	⊥	\perp
∝	\propto						

Tabelul 7.7: Operatori relaţionali

≮	\not<	≯	\not>	≠	\not=
≰	\not\leq	≱	\not\geq	≢	\not\equiv
⊀	\not\prec	⊁	\not\succ	≁	\not\sim
⋠	\not\preceq	⋡	\not\succeq	≄	\not\simeq
⊄	\not\subset	⊅	\not\supset	≉	\not\approx
⊈	\not\subseteq	⊉	\not\supseteq	≇	\not\cong
⋢	\not\sqsubseteq	⋣	\not\sqsupseteq	≭	\not\asymp

Tabelul 7.8: Operatori de negaţie

Tabelele **7.6÷7.8** conţin alte simboluri, numite operatori mici, la care comanda `\limits` nu are nici un efect.

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleft...`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleft...`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`			$\leadsto$	`\leadsto`

Tabelul 7.9: Simboluri de tip săgeată

$\neq$	`\ne`, `\neq` sau `\not=`	$\leq$	`\le` sau `\leq`
$\geq$	`\ge` sau `\geq`	$\{$	`\{` sau `\lbrace`
$\}$	`\}` sau `\rbrace`	$\to$	`\to` sau `\rightarrow`
$\leftarrow$	`\gets` sau `\leftarrow`	$\ni$	`\owns` sau `\ni`
$\wedge$	`\land` sau `\wedge`	$\vee$	`\lor` sau `\vee`
$\neg$	`\lnot` sau `\neg`	$\vert$	`\vert` sau `|`
$\Vert$	`\Vert` sau `\\|`		

Tabelul 7.10: Comenzi sinonime

Tabelul **7.10** conţine seturi de comenzi care au acelaşi efect.

Literele alfabetului grecesc sunt disponibile, atât ca litere mici cât şi ca majuscule, numai în mod matematic (vezi tabelele **7.11** şi **7.12**).

7.2 Spaţii în mod matematic

În mod matematic există următoarele comenzi de spaţiere:

- `\,` spaţiu îngust;
- `\:` spaţiu mediu;

α	\alpha	ι	\iota	ϱ	\varrho
β	\beta	κ	\kappa	σ	\sigma
γ	\gamma	λ	\lambda	ς	\varsigma
δ	\delta	μ	\mu	τ	\tau
ϵ	\epsilon	ν	\nu	υ	\upsilon
ε	\varepsilon	ξ	\xi	ϕ	\phi
ζ	\zeta	o	o	φ	\varphi
η	\eta	π	\pi	χ	\chi
θ	\theta	ϖ	\varpi	ψ	\psi
ϑ	\vartheta	ρ	\rho	ω	\omega

Tabelul 7.11: Literele mici ale alfabetului grecesc

Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

Tabelul 7.12: Literele mari ale alfabetului grecesc

- `\;` spaţiu mare;
- `\!` spaţiu negativ îngust;
- `\quad` un spaţiu egal cu de 6 ori spaţiul lăsat de `\,`;
- `\qquad` un spaţiu egal cu de 10 ori spaţiul lăsat de `\,`;

Exemplu cu comenzile de spaţiere (comanda de spaţiere va fi pusă între x şi y din expresia $axyz$):

spaţiu normal: $axyz$
spaţiu îngust: $ax\, yz$
spaţiu mediu: $ax\: yz$
spaţiu mare: $ax\; yz$
spaţiu negativ îngust: $ax\! yz$
spaţiu quad: $ax\quad yz$
spaţiu qquad: $ax\qquad yz$

```
\[ \begin{array}{rl}
\mbox{spaţiu normal:}&axyz\\
\mbox{spaţiu îngust:}&ax\, yz\\
\mbox{spaţiu mediu:}&ax\: yz\\
\mbox{spaţiu mare:}&ax\; yz\\
\mbox{spaţiu negativ îngust:}&
ax\! yz  \\
\mbox{spaţiu quad:}&ax\quad yz\\
\mbox{spaţiu qquad:}&ax\qquad yz
\end{array}\]
```

Comenzile de spaţiere sunt deosebit de utile:

frumos: $\int_0^a \int_0^b dx\,dy = ab \quad \forall a,b \in \Re$

```
frumos:
$$\int\limits_0^a\!\!\int\limits_0^bdx\,dy
=ab\quad\forall a,b\in\Re$$
ur\^at:
$$\int\limits_0^a\int\limits_0^bdxdy=ab
\forall a,b\in\Re$$
```

urât: $\int_0^a \int_0^b dxdy = ab \forall a,b \in \Re$

7.3 Tipuri şi dimensiuni de caractere în mod matematic

Pe lângă tipurile `\cal` (descris deja) şi `\mit` (mathitalic)—tip implicit în mod matematic— mai există şi tipul `\boldmath` care permite scrierea de caractere îngroşate. Totuşi această comandă nu poate fi folosită în mod matematic decît ca argument al comenzii `\mbox`. Indicii nu vor fi scrişi îngroşat. Opusă comenzii `\boldmath` este comanda `\unboldmath` care revine la scrierea cu tipul implicit.

Atunci când se scriu formule, TEX foloseşte următoarele 4 stiluri:

- `display` simboluri normale în formule separate de text;
- `text` simboluri folosite în scrierea formulelor *în text*;
- `script` simboluri folosite pentru indicii superiori şi inferiori;
- `scriptscript` simboluri pentru indici ai indicilor de orice nivel;

Cu ajutorul comenzilor `\displaystyle`, `\textstyle`, `\scriptstyle` şi respectiv `\scriptscriptstyle`, a căror semnificaţie rezultă din cele de mai sus, TEX poate fi forţat să folosească unul din aceste stiluri.

Exemplu:

$$\stackrel{a}{b+c},\quad \stackrel{\textstyle a}{b+c},\quad \stackrel{\textstyle\scriptstyle a}{\scriptstyle b+c}.$$

```
\[
\stackrel{a}{b+c},\quad
\stackrel{\textstyle a}{b+c},\quad
\stackrel{\textstyle a}{
   \scriptstyle b+c}.
\]
```

Pe lângă aceste stiluri vom putea folosi şi altele încărcate cu comanda `\load{`*dim*`}{`*stil*`}`, care are ca efect încărcarea stilului *stil* cu dimensiunea *dim*. Comanda `\load` trebuie să apară înaintea folosirii pentru prima dată a fontului în mod matematic şi nu are voie să fie inclusă între acolade sau să fie în interiorul cadrurilor. Totuşi alegerea fonturilor este restrânsă la următoarele două reguli:

a) stilul *script* şi *scriptscript* poate folosi numai fonturi preîncărcate (vezi tabelul **6.1** de la pagina 63);

b) o combinaţie stil-dimensiune corespunzătoare unui font încărcat la cerere foloseşte acelaşi font pentru toate stilurile matematice. Ceea ce poate duce (pentru anumite combinaţii stil-dimensiune), uneori, la scrierea greşită a indicilor (ca dimensiune);

Capitolul 8

Modul LR

Modul LR (*Left to Right*) este un mod special, în care TEX tipăreşte text de la stânga la dreapta fără să facă verificări asupra încadrării în spaţiul disponibil. În acest mod TEX nu va trece niciodată la rând nou.

Atunci când TEX aranjează texte în pagină, el lucrează cu „cutii" despre care cunoaşte doar lăţimea şi înălţimea. În această fază, el tratează la fel o literă şi, de exemplu, un cadru `minipage`, singurul lucru care le diferenţiază fiind dimensiunile cutiei în care ele încap. De regulă un text procesat în mod LR este introdus într-o cutie, urmând ca la ieşirea din modul LR această cutie să fie aşezată în pagină. Există tot timpul riscul ca un text procesat în mod LR să depăşească lăţimea paginii.

Comenzile care culeg text în acest mod pot apărea la rândul lor în orice mod (paragraf, matematic sau LR):

- `\mbox{`*text*`}` are ca efect scrierea lui *text* în mod LR;
- `\makebox[`*dim*`][`*loc*`]{`*text*`}` creează o cutie de lăţime *dim* punând textul *text* în interiorul acestuia în funcţie de valorile lui *loc* (*loc* poate fi `r`,`l` sau `c` cu semnificaţia cunoscută deja de la cadrul `tabular`);
- `\fbox{`*text*`}` are aceeaşi semnificaţie ca şi `\mbox` doar că, de data aceasta textul va fi încadrat într-un dreptunghi;
- `\framebox[`*dim*`][`*loc*`]{`*text*`}` are efect similar cu `\makebox` dar desenează efectiv dreptunghiul;
- `\raisebox{`*dy*`}{`*text*`}` ridică sau coboară textul *text* cu cantitatea *dy* după cum *dy*>0 sau *dy*<0;

- `\raisebox{`*dy*`}[`*înălţime*`][`*adâncime*`]{`*text*`}` ca mai sus dar textul *text* se va extinde cu *înălţime* deasupra rândului şi cu *adâncime* sub rând;

Faptul că aceste comenzi pot apărea în orice mod oferă un avantaj mare: Dacă un anumit simbol matematic apare adesea când în mod matematic, când în mod paragraf, şi am vrea să definim un macrou care să-l producă, punem simbolul în argumentul unei comenzi `\mbox`, şi astfel vom putea folosi comanda în orice mod:

... $g_{\cdot i}^{j}$ se defineşte:

$$g_{\cdot i}^{j} = \ldots$$

```
\newcommand{\gij}
{\mbox{$g_{\cdot i}^{j\cdot}$}}

... \gij\ se defineşte:
$$\gij=\ldots$$
```

Stilul este controlat de următorii parametri:

- `\fboxrule` reprezintă lăţimea liniei trasate de `\fbox` şi `\framebox`;
- `\fboxsep` reprezintă spaţiul dintre marginea unui cadru şi conţinutul său. Nu se aplică în cazul celor desenate in interiorul cadrului `picture`;

LaTeX oferă o modalitate de a salva o „cutie“ astfel încât ea poate fi folosită de oricâte ori fără a se mai procesa conţinutul ei, ceea ce înseamnă economie de timp la procesarea documentului, mai ales la folosirea unor desene mai complicate. Comezile folosite sunt:

- `\newsavebox{\`*cda*`}` declară `\`*cda* ca fiind un *buffer* pentru salvarea unei „cutii“.
- `\sbox{\`*cda*`}{`*text*`}`, `\savebox{\`*cda*`}[`*dim*`][`*loc*`]{`*text*`}` acţionează asemănător cu `\mbox` respectiv `\makebox`, dar în loc să tipărească conţinutul îl salvează în *buffer*-ul `\`*cda* declarat deja cu `\newsavebox`;

Un cadru salvat cu comenzile de mai sus poate fi tipărit cu comanda `\usebox{\`*cda*`}`.

Semnul este deja procesat şi va folosită reprezentarea cutiei în care se încadrează el: .

```
\newsavebox{\cutie}
.....................
\savebox{\cutie}{
\begin{picture}(10,10)
\put(0,0){\line(1,0){10}}
\put(0,0){\line(0,1){10}}
\put(10,10){\line(0,-1){10}}
\put(10,10){\line(-1,0){10}}
\end{picture}}
Semnul \usebox{\cutie} este deja procesat
\c si va folosită reprezentarea cutiei
\^\i n care se \^\i ncadreaz\u a el:
\usebox{\cutie}.
```

Capitolul 9

Variabile interne LaTeX

9.1 Lungimi

LaTeX lucrează cu lungimi de oricâte ori trebuie să dimensioneze un obiect. De exemplu, primul parametru al comenzii `\parbox` trebuie să fie o lungime. Putem transmite aici o construcţie de genul

$$< \textit{parte întreagă} >< \textit{patre zecimală} >< \textit{unitate de măsură} >$$

dar putem transmite şi o variabilă de tip lungime. O astfel de variabilă poate fi una din cele predefinite în LaTeX (vezi şi secţiunile următoare) sau una definită de utilizator. Putem folosi următoarele comenzi pentru modificarea valorilor variabilelor de tip lungime:

- `\setlength{\cda}{val}` setează comanda de lungime `\cda` la valoarea *val*;
- `\addtolength{\cda}{val}` adaugă comenzii de lungime `\cda` valoare *val*;
- `\settowidth{\cda}{text}` setează comanda de lungime `\cda` la valoarea lăţimii textului *text*;
- `\stretch{număr}` o lungime naturală egală cu 0 şi de *număr* de ori întinderea lui `\fill`;
- `\newlength{\cda}` defineşte `\cda` ca fiind o nouă variabilă de tip lungime;

Unităţile de măsură valide ale lungimilor sunt:

1in(inch)	=	25.4mm
1in(inch)	=	72.27pt
1em	=	aproximativ lăţimea în pt a literei `m`
1ex	=	aproximativ înălţimea în pt a literei `x`
1cm	=	10mm
1mm	=	2.845276pt
1pc(Picas)	=	12pt
1pt(Point)	=	$\frac{1}{72.27}$inches
1bpt(Big point)	=	$\frac{1}{72}$inches
`\fill`	=	spaţiu de lungime variabilă ($0 \rightarrow \infty$)

Unele dintre cele mai importante variabile de tip lungime ale căror valori sunt dependente de stilul documentului sunt:

- `\parindent` lungimea alineatului la începutul unui paragraf;
- `\textwidth` lăţimea textului pe pagină;
- `\textheight` lungimea textului pe pagină;
- `\parskip` spaţiul vertical suplimentar dintre ultima linie a paragrafului şi prima linie a paragarfului următor;
- `\baselineskip` spaţiul dintre sfârşitul unui rând şi sfârşitul rândului următor;

Un mic exemplu: dacă nu dorim alineate la început de paragraf, putem da comanda

```
\setlength{\parindent}{0cm}
```

la începutul documentului.

9.1.1 Lungimi pentru controlul *layout*-ului

Stilurile `article` şi `report`, fără opţiunea `twoside`

În figura **9.2** este dată o reprezentare schematică a unei coli de hârtie, împreună cu variabilele de tip lungime cu care lucrează LaTeX pentru a controla designul ei. Schema este valabilă atât pentru pagini cu număr par cât şi pentru cele cu număr impar dacă nu se foloseşte opţiunea `twoside` la `\documentstyle`, sau dacă stilul nu este `book`. Astfel, o pagină se va compune de sus în jos, respectiv de la stânga la dreapta din:

Pe verticală: 1 inch, `\topmargin`, `\headheight`, `\headsep`, `\textheight`, `\footskip`, restul paginii.
Pe orizontală: 1 inch, `\oddsidemargin` sau `\evensidemargin`, `\textwidth`, `\marginparsep`, `\marginparwidth`, restul paginii.

Singura diferenţă dintre paginile pare şi impare o constituie lungimea numită `\oddsidemargin` respectiv `\evensidemargin`, dar care au aceeaşi valoare. Pentru toate paginile, notele marginale sunt puse în marginea dreaptă a paginii.

Stilul `book` sau orice stil cu opţiunea `twoside`

Dacă se face culegere diferenţiată a textului pe pagini pare şi impare, ceea ce este implicit la stilul `book`, sau la opţiunea `twoside`, paginile pare au o componenţă diferită de cele impare pentru care rămâne valabilă schema din figura **9.2**. În figura **9.1** e dată schema unei pagini pare. Se observă că notele marginale sunt puse în partea stângă.

Putem utiliza comenzile:

```
\addtolength{\textheight}{2cm}
\addtolength{\topmargin}{-1cm}
```

pentru a lungi textul de pe pagină.

Comanda `\baselinestretch` este singura dintre toate secvenţele de control din cele două figuri care nu este o lungime. Modificarea ei se va face cu comanda `\renewcommand`. Implicit ea are valoarea 1. Pentru ca textul să fie cules la "un rând şi jumătate", se dă comanda

```
\renewcommand{\baselinestretch}{1.3}
```

respectiv, pentru culegere "la două rânduri":

```
\renewcommand{\baselinestretch}{1.6}
```

9.1.2 Designul unei pagini

Fiecare stil de document are predefinit un design specific pentru pagini. Putem opta însă pentru altul dacă acesta nu ne convine.

Fiecare pagină este compusă din antet, corp şi subsol. Dimensiunile acestor trei elemente sunt determinate de stilul documentului iar conţinutul antetului şi al subsolului este determinat de stilul paginii. Stilul paginii poate fi modificat cu comanda:

`\pagestyle{`*stil*`}`

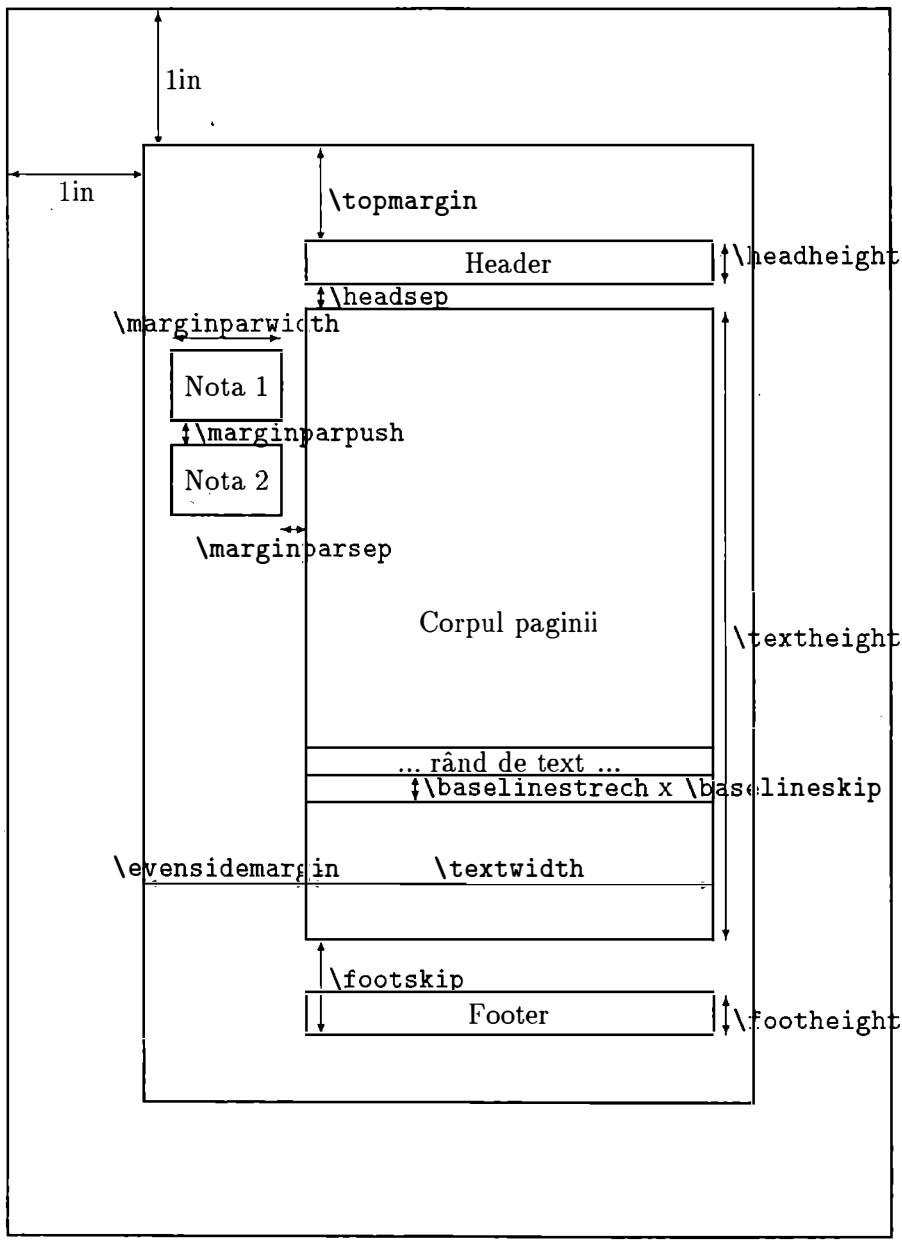


Figura 9.1: Componenţa unei pagini pare dacă se face culegere diferenţiată a textului pe pagini pare şi impare.

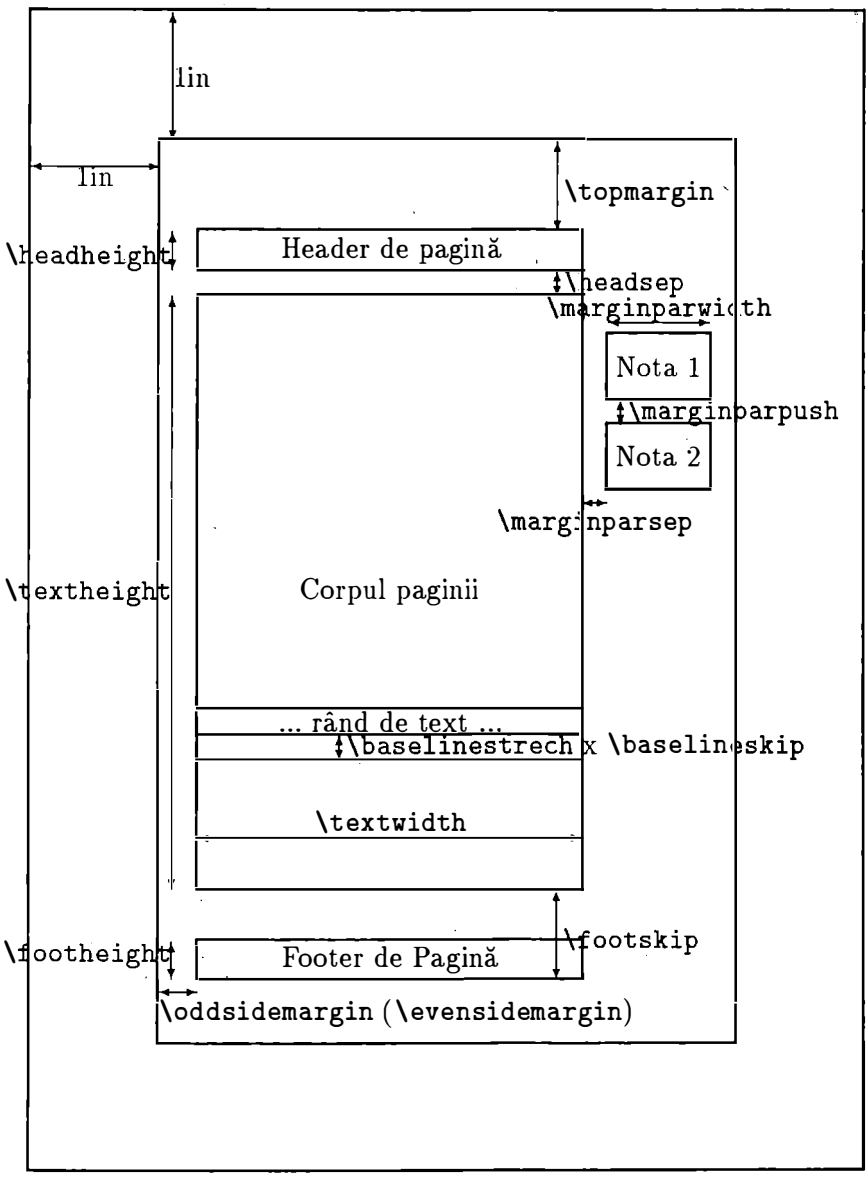


Figura 9.2: Componenţa unei pagini impare pentru orice stil de document (şi a unei pagini pare dacă nu se face culegere diferenţiată a textului pe pagini pare şi impare).

pentru toate paginile, sau cu comanda

`\thispagestyle{`*stil*`}`

pentru pagina curentă. *stil* poate fi unul din următoarele stiluri de pagină:

`plain` Este stilul implicit. Antetul va fi vid, iar în subsol va apărea doar numărul paginii.

`empty` Antetul şi subsolul sunt vide.

`headings` Antetul conţine informaţii secţionale, determinate de stilul documentului, şi numărul paginii. Subsolul este vid.

`myheadings` Antetul va conţine informaţii stabilite de utilizator şi numărul paginii. Subsolul este vid.

Dacă se foloseşte stilul `myheadings`, informaţiile pentru antete sunt stabilite cu ajutorul comenzilor:

`\markboth{`*antet stânga*`}{`*antet dreapta*`}`

`\markright{`*antet dreapta*`}`

unde *antet stânga* nu se va folosi decât dacă s-a optat pentru `twoside` în lista de opţiuni ai comenzii `\documentstyle`, caz în care se va folosi pentru antetele paginilor pare.

Felul în care se face numerotarea paginilor se poate stabili cu comanda `\pagenumbering`:

`\pagenumbering{`*stil*`}`

unde *stil* poate fi unul din următoarele cuvinte:

`arabic` Numere arabe (1, 2, 3 ...).

`roman` Numere romane mici (i, ii, iii, ...).

`Roman` Numere roamne mari (I, II, III, ...).

`alph` Litere mici (a, b, c, ...).

`Alph` Litere mari (A, B, C, ...).

Avem posibilitatea tipăririi pe două coloane a întregului document specificând `twocolumn` în lista de opţiuni a comenzii `\documentstyle`, dar ne stau la dispoziţie şi comenzile:

`\twocolumn[`*text*`]`

care determină trecerea la pagină nouă, tipărirea lui *text* (dacă `[`*text*`]` este prezent) pe toată lăţimea paginii şi culegerea textului ce urmează după comandă pe două coloane, şi comanda

`\onecolumn`

care determină trecerea la pagină nouă şi culegerea textului ce urmează comenzii pe toată lăţimea paginii.

9.2 Variabile întregi

Pentru o serie de operaţii cum ar fi numerotarea paginilor, ecuaţiilor, secţiunilor etc., LaTeX foloseşte contoare. Contoarele predefinite sunt următoarele:

- `part` se foloseşte la numeroatrea părţilor în cazul stilului `book` şi `report`;
- `chapter` se foloseşte la numerotarea capitolelor în cazul stilului `book` şi `report`;
- `section`, `subsection`, `subsubsection` folosite la numerotarea secţiunilor, subsecţiunilor şi a subsubsecţiunilor;
- `paragraph`, `subparagraph` folosite la numerotarea paragrafelor şi a subparagrafelor;
- `page` folosit la numerotarea paginilor;
- `equation` folosit la numeroatarea ecuaţiilor;
- `figure`, `table` folosite la numerotarea figurilor respectiv numerotarea tabelelor;
- `footnote`, `mpfootnote` folosite la numerotarea notelor de subsol şi a notelor de subsol dintr-un cadru `minipage`;
- `enumi`, `enumii`, `enumiii`, `enumiv` folosite la numerotarea intrărilor într-un cadru `enumerate` pentru diferite niveluri de imbricare (maximum 4);

Asupra contoarelor se poate acţiona cu ajutorul următoarelor comenzi:

- `\setcounter{`*contor*`}{`*val*`}` setează contorul *contor* la valoarea *val*;
- `\addtocounter{`*contor*`}{`*val*`}` adaugă la valoarea curentă a contorului *contor* valoarea *val*;
- `\value{`*contor*`}` reprezintă valoarea contorului *contor* pentru a fi folosită ca parametru în alte comenzi;
- `\the`*contor* tipăreşte valoarea contorului *contor* (de exemplu comanda `\thepage` tipăreşte valoarea curentă a numărului de pagină);

Valorile contoarelor pot fi tipărite în cinci stiluri diferite. În continuare dăm comenzile folosite pentru aceasta:

- `\arabic{`*contor*`}` tipăreşte cifra arabă asociată contorului *contor*;
- `\roman{`*contor*`}` tipăreşte cifra romană asociată valorii contorului *contor*;
- `\Roman{`*contor*`}` ca mai sus dar cifra romană este scrisă cu majuscule;
- `\alph{`*contor*`}` tipăreşte litera mică din alfabet a cărei poziţie corespunde cu valoarea contorului (maximum 26 corespunzător literei z);
- `\Alph{`*contor*`}` tipăreşte majuscula din alfabet a cărei poziţie corespunde cu valoarea contorului (maximum 26 corespunzător literei Z);
- `\fnsymbol{`*contor*`}` produce caracterele *, †, ‡, §, ¶, ||, **, ††, ‡‡. Poate fi folosit doar în mod matematic iar valoarea contorului *contor* trebuie să fie în intervalul $1 \div 9$.

Definirea de contoare noi se face cu comanda:

`\newcounter{`*contornou*`}[`*contorextern*`]`

care defineşte un nou contor *contornou*. Forma opţională are ca efect definirea contorului în interiorul lui *contorextern*. Aceasta are ca efect aducerea la zero a lui *contornou* de fiecare dată când valoarea lui *contorextern* se modifică.

Comenzile `\stepcounter{`*contor*`}` şi `\refstepcounter{`*contor*`}` au ca efect incrementarea cu 1 a valorii contorului *contor* şi poziţionarea pe 0 a tuturor contoarelor definite în interiorul acestui contor (vezi comanda `\newcounter`). În plus, comanda `\refstepcounter` declară valoarea curentă atribuită cu `label` ca fiind textul generat de `\the`*contor*.

9.3 Parametrii de stil

Uneori atunci când se doreşte modificarea stilului în care sunt scrise anumite obiecte trebuie modificaţi anumiţi parametri specifici stilului, aşa numiţii *parametrii de stil*. Modificarea acestor parametri trebuie să se facă cu grijă şi în general nu este o acţiune încurajată deoarece valorile acestor parametri au fost stabilite respectând anumite reguli tipografice (în vigoare în SUA). Parametrii de stil pot fi lungimi, contoare sau secvenţe de control.

9.3.1 Ecuaţii matematice

Următoarele lungimi controlează formatul ecuaţiilor matematice:

- `\mathindent` reprezintă spaţiul de la marginea unui paragraf până la începutul unei formule (numai când s-a ales opţiunea `fleqn`);
- `\jot` reprezintă spaţiul vertical adăugat între două linii din cadrul `eqnarray` şi `eqnarray*`;
- `\abovedisplayskip` reprezintă spaţiul adăugat deasupra unei formule mai lungi în cazul în care nu s-a optat pentru `fleqn` (caz în care se foloseşte ca spaţiu valoarea lui `\topsep`);
- `\belowdisplayskip` reprezintă spaţiul adăugat sub o formulă mai lungă cu excepţia stilului `fleqn` (caz în care se foloseşte ca spaţiu valoarea lui `\topsep`);
- `\abovedisplayshortskip` reprezintă spaţiul adăugat deasupra unei formule mai scurte dacă comenzii `\documentstyle` nu i s-a transmis opţiunea `fleqn` (caz în care se foloseşte ca spaţiu valoarea lui `\topsep`);
- `\belowdisplayshortskip` reprezintă spaţiul adăugat sub o formulă mai scurtă cu excepţia stilului `fleqn` (caz în care se foloseşte ca spaţiu valoarea lui `\topsep`);

Formatul numărului ecuaţiei este controlat de secvenţa de control `\theequation`, ce poate fi eventual redefinită. De exemplu, pentru stilul `book` `\theequation` este predefinit ca `{\thechapter.\arabic{equation}}`.

$$x_n = x_{n-1} + 1 \qquad n > 1 \qquad (9.1a)$$

$$x_{n+1} = x_n + 1 \qquad n > 0 \qquad (9.1b)$$

Ecuaţiile 9. 1a şi 9. 1b sunt echivalente.

```
{\renewcommand{\theequation}{\thechapter.
\arabic{equation}a}\begin{equation}
\label{Sirn}x_n=x_{n-1}+1 \qquad n>1
\end{equation}\addtocounter{equation}{-1}
\renewcommand{\theequation}{\thechapter.
\arabic{equation}b}\begin{equation}
\label{Sirn+1}x_{n+1}=x_n+1 \qquad n>0
\end{equation}}
Ecuaţiile \ref{Sirn} şi \ref{Sirn+1}
sunt echivalente.
```

De remarcat includerea primei părţi a exemplului într-un grup, ceea ce face ca în afara lui să se folosească definiţia implicită pentru `\theequation`, deoarece definiţiile noastre date cu `\renewcommand` sunt valabile doar în grupul în care au fost făcute.

9.3.2 Cadrurile `array` şi `tabular`

Formatul acestor două cadruri este controlat de următoarele lungimi:

- `\arraycolsep` reprezintă jumătatea lăţimii spaţiului orizontal dintre două coloane succesive într-un cadru `array`;
- `\tabcolsep` reprezintă jumătatea lăţimii spaţiului orizontal dintre două coloane succesive într-un cadru `tabular` sau `\tabular*`;
- `\arrayrulewidth` reprezintă lăţimea unei linii create cu `|`, `\hline`, `\cline` sau `\vline`;
- `\doublerulesep` reprezintă lăţimea spaţiului dintre două linii succesive create cu caracterele `||`;
- `\arraystretch` controlează spaţiul dintre rânduri. Spaţiul dintre rânduri este multiplicat cu această valoare (implicit valoarea este 1);

9.3.3 Cadrurile `table` şi `figure`

În continuare vom folosi termenul *flotant* cu care vom desemna obiectele create cu cele două cadruri[1].

- `topnumber` este un contor a cărui valoare reprezintă numărul maxim de flotanţi permişi în partea de sus a paginii. Valoare implicită: 2.

[1] În limba engleză termenul folosit este *float*

- `\topfraction` este o secvenţă de control ce reprezintă fracţiunea maximă dintr-o pagină care poate fi ocupată de flotanţi în partea de sus. Valoare implicită: 0.7.

- `bottomnumber` este un contor a cărui valoare reprezintă numărul maxim de flotanţi permişi în partea de jos a paginii. Valoare implicită: 1.

- `\bottomfraction` este o secvenţă de control ce reprezintă fracţiunea maximă dintr-o pagină care poate fi ocupată de flotanţi în partea de jos. Valoare implicită: 0.3.

- `totalnumber` este un contor a cărui valoare reprezintă numărul maxim de flotanţi permişi pe o pagină. Valoare implicită: 3.

- `\textfraction` este o secvenţă de control ce reprezintă fracţiunea minimă dintr-o pagină care va trebui ocupată de text. Valoare implicită: 0.2.

- `\floatpagefraction` este o secvenţă de control ce reprezintă fracţiunea maximă dintr-o pagină care va putea fi ocupată de flotanţi. Valoare implicită: 0.5.

- `dbltopnumber` este un contor a cărui valoare reprezintă numărul maxim de flotanţi permişi în partea de sus a paginii în cazul stilului pe două coloane (`twocolumn`). Valoare implicită: 2.

- `\dbltopfraction` este analogul lui `\topfraction` în cazul stilului pe două coloane (`twocolumn`). Valoare implicită: 0.7.

- `\dblfloatpagefraction` este analogul lui `\floatpagefraction` în cazul stilului pe două coloane (`twocolumn`). Valoare implicită: 0.5.

- `\floatsep` este o lungime ce indică spaţiul vertical suplimentar adăugat între doi flotanţi.

- `\textfloatsep` este o lungime ce reprezintă spaţiul vertical suplimentar adăugat între text şi flotanţi.

- `\intextsep` este o lungime ce reprezintă spaţiul vertical suplimentar adăugat deasupra şi dedesubtul unui flotant când acesta se află în mijlocul textului;

- `\dblfloatsep` este analogul lui `\floatsep` în cazul stilului pe două coloane (`twocolumn`).
- `\dbltextfloatsep` este analogul lui `\textfloatsep` în cazul stilului pe două coloane (`twocolumn`).

Capitolul 10

Erori şi avertismente

Mai ales la început, dar chiar şi după o lungă experienţă în folosirea programului LaTeX, se întâmplă să facem tot felul de greşeli. LaTeX nu poate verifica gramatical textul, deci nu vom vorbi aici despre astfel de erori, ci despre erori în folosirea comenzilor LaTeX.

Cel mai adesea se întâmplă să greşim numele unei comenzi. De exemplu să presupunem că în fişierul sursă, linia 4 conţine următorul text:

```
\tableofcontents \listofigures \listoftables
```

La procesarea fişierului, se va tipări pe ecran mesajul:

```
.........
! Undefined control sequence.
l.4 \tableofcontents \listofigures
                                    \listoftables

?
```

şi LaTeX aşteaptă o reacţie din partea noastră. Dacă apăsăm <ENTER>, comanda ce nu a fost înţeleasă este ignorată şi se continuă procesarea. În cazul nostru, dacă nu mai există greşeli, rezultatul va fi că nu s-a generat lista de figuri, deoarece comanda corectă este `\listoffigures`.

Linia care începe cu semnul de exclamare este mesajul de eroare. Următoarea linie ne spune că eroarea a fost întâlnită în linia 4 a fişierului sursă (dacă lucraţi cu fişiere incluse, trebuie să determinaţi mai întâi în ce fişier e eroarea), pe urmă e reprodusă această linie, ruptă imediat după comanda eronată.

Avem de ales între 5 acţiuni în momentul apariţiei unei erori:

- Apăsarea tastei <ENTER> pentru a ignora această eroare, dar în multe cazuri ignorarea unei erori duce la apariţia mai multor alte erori care depind de comanda eronată detectată. Deci nu vă grăbiţi să tastaţi <ENTER>.
- Apăsarea tastei I urmată de <ENTER> pentru a insera de la tastatură textul corect. Folosiţi însă cu prudenţă această comandă, deoarece ea nu face modificarea decât pe moment, în fişierul sursă eroarea rămâne.
- Apăsarea tastei Q urmată de <ENTER> pentru procesarea în continuare, cu ignorarea eventualelor comenzi eronate ce ar mai putea apărea.
- Apăsarea tastei X urmată de <ENTER> pentru a renunţa la procesarea în continuare. Este soluţia ce o recomandăm noi, deoarece astfel putem corecta această eroare şi reprocesa documentul, eliminând erorile una câte una.
- Apăsarea tastei H urmată de <ENTER> pentru a obţine un scurt mesaj de ajutor (*help*).

Eroarea de mai sus a fost descoperită de TEX, nu de LATEX. Dacă LATEX descoperă o eroare, mesajul de eroare începe cu `LaTeX error.`. Dar pentru utilizator e mai puţin important dacă e vorba de o eroare LATEX sau TEX, important este să fie eliminată, lucru care nu este întotdeauna atât de simplu ca la exemplul de mai sus, unde nu avem decât de răspuns X <ENTER>, de corectat fişierul sursă, şi reprocesat documentul.

10.1 Mesaje de eroare

În continuare, vom da mesajul de eroare şi o scurtă explicaţie despre el. În paranteză e specificat cine raportează eroarea.

`! Bad math environment delimiter.` (LATEX)
Se încearcă intrarea din mod matematic în mod matematic sau ieşirea din mod matematic fără a fi în el.

`! Can be used only in preamble.` (LATEX)
Se foloseşte una din comenzile: `\documentstyle`, `\nofiles`, `\includeonly`, `\makeindex` sau `\makeglosary` *după* `\begin{document}`.

`! Comand name ... already used.` (LATEX)
Se încearcă folosirea unei comenzi `\new...` pentru declararea unei comenzi, lungimi, unui contor sau cadru deja existent.

`! Extra alignement tab has been changed to \cr.`(TEX)
Prea multe caractere & pe o linie a unui cadru `tabular` sau `array`. Probabil s-a uitat o comandă `\\` între rândul de mai sus şi cel pe care s-a raportat eroarea.

`! Extra } or forgotten $.`(TEX)
Acoladele sau delimitatorii modului matematic nu sunt perechi.

`! Illegal parameter number in definition of ... .`(TEX)
Caracterul special `#` folosit impropriu la definirea unei comenzi cu parametri.

`! Misplaced alignament tab character &.`(TEX)
Caracterul `&` folosit impropriu (el poate fi folosit doar ca tabulator în cadrurile `tabular` şi `array`).

`! Missing number, treated as zero.`(TEX)
S-a uitat un argument ce trebuia să fie un număr.

`! Missing {.`
`! Missing }.`(TEX)
Probabil acoladele nu sunt perechi. Cauza erorii poate fi mult mai sus decât locul unde a fost detectată eroarea.

`! Missing $ inserted.`(TEX)
Probabil se încearcă folosirea unei comenzi matematice fără a fi în mod matematic. De asemenea, eroarea aceasta poate fi raportată dacă se întâlneşte o linie goală în mod matematic.

`! Not in outer par mode.`(LATEX)
S-a pus un cadru mobil sau o comandă `\marginpar` într-un `parbox`, de exemplu într-un cadru `minipage`, sau într-un alt cadru mobil), sau în mod matematic.

`! Pargraph ended before ... was complete.`(TEX)
Argumentul unei comenzi nu poate conţine mai multe paragrafe (linie goală sau comanda `\par`).

`! Something's wrong--perhaps a missing \item.`(LATEX)
De regulă apare în liste dacă se uită o comandă `\item`. Apare şi când se uită argumentul cadrului `thebibliography`.

`! Too deeply nested.`(LATEX)
Unele variante de LATEX nu permit o imbricare mai mare de 4 a listelor.

`! Too many unprocessed floats.`(LATEX)
Se folosesc prea multe cadruri mobile. Acest lucru se poate întâmpla mai

ales când un cadru mobil nu poate fi aşezat pe pagină dintr-un motiv sau altul, astfel încât nici următoarele nu pot fi aşezate, din moment ce LaTeX nu modifică niciodată ordinea lor. Adesea e suficient să se dea parametrul opţional p cadrului care cauzează problema (pentru a-l identifica, uitaţi-vă la rezultatul procesării, căutând ultimul cadru mobil. Următorul este probabil cel ce a cauzat eroarea.

`! Use of ... doesn't match its definition`(TeX)
Folosirea comenzii (sau a cadrului) ... nu corespunde definiţiei ei (lui).

10.2 Mesaje de avertisment

Mesajele de avertisment date de LaTeX încep toate cu "`LaTeX Warning:`".

`LaTeX Warning: Citation '...' on page ... undefined.`
De regulă apare la prima procesare a unui document ce foloseşte citări din bibliografie. Dacă şi la a doua procesare avertismentul rămâne, se foloseşte o comandă `\cite` cu o etichetă nedeclarată printr-o comandă `\bibitem`.

`LaTeX Warning: Label '...' multiply defined.`
Două sau mai multe comenzi `\label` sau `\bibitem` au acelaşi argument (etichetele trebuie să fie unice într-un document!).

`LaTeX Warning: Marginpar on page ... moved.`
O notă marginală începe altundeva decât în dreptul rândului de text în care apare comanda, deoarece acolo nu mai era loc.

`Overfull \hbox ...`(TeX)
Lăţimea textului depăşeşte lăţimea permisă în acel loc. În mod paragraf, problema e de regulă faptul că LaTeX nu poate face despărţirea în silabe a ultimului cuvânt de pe linia unde a apărut mesajul. Dacă mesajul de avertisment conţine şi mesajul `... while \output is active.`, atunci textul care a cauzat avertismentul este probabil un parametru mobil (un antet, un parametru al unei comenzi `\caption` etc.).

`Overfull \vbox ...`(TeX)
Înălţimea textului de pe o pagină depăşeşte lungimea permisă.

`LaTeX Warning: Reference '...' on page ... undefined.`
Este dat la întâlnirea unei comenzi `\ref` pentru care în fişierul `.aux` nu se găseşte comanda `\newlabel` aferentă. Este cazul procesării pentru prima

dată a fişierului, deoarece abia atunci se scrie fişierul `.aux`. Dacă după reprocesare mesajul de avertisment e din nou afişat, atunci aţi uitat să puneţi în document comada `\label` pentru eticheta `'...'`.

`Underfull \hbox ...`(TEX)
Spaţiile dintre cuvinte sunt prea mari, de regulă apare când se foloseşte comanda `\sloppy` sau într-un cadru `sloppypar`.

`LaTeX Warning: ... in math mode.`
Apare dacă o comandă ce nu este permisă în mod matematic apare totuşi în acest mod.

•

Încercăm să dăm aici o strategie care nu este chiar cea mai rapidă, dar care este mai simplă, pentru soluţionarea erorilor mai complexe. Pe măsură ce utilizatorul va căpăta mai multă experienţă, îşi va forma propria lui părere şi va hotărâ singur în ce fel să rezolve erorile.

Recomandăm întreruperea procesării la apariţia fiecărei erori, şi corectarea ei. Dacă dorim să vedem mai întâi toate erorile, riscăm să ne încurcăm în ele pe de o parte, iar pe de altă parte, unele ar putea fi datorate unei erori precedente şi ar dispărea odată cu eliminarea acesteia.

Uneori nu este evidentă greşeala care a produs eroarea. Cauza unei erori se va căuta din linia raportată de LATEX în sus. Dacă nu o găsiţi, faceţi o copie a fişierului pe care o puneţi la loc sigur, şi ştergeţi din el partea de după linia la care a ajuns LATEX cu procesarea în momentul raportării erorii.

Înjumătăţiţi acum partea ce a rămas şi ştergeţi prima parte, dar cu mare atenţie, să nu ştergeţi şi declaraţii sau alte lucruri de care depinde a doua jumătate. Procesaţi ce a rămas şi reluaţi operaţia până când eroarea nu mai este raportată. Acest lucru înseamnă că eroarea se afla în ultima jumătate ştersă, şi veţi relua căutarea pentru aceasta.

Anexa A

Lucrul cu fişiere incluse

Atunci când scriem documente lungi, este avantajos să împărţim textul sursă în mai multe fişiere de dimensiuni mai reduse. Am prezentat deja comanda `\input` în captiolul 5, unde am propus scrierea macrodefiniţiilor mai des utilizate într-un fişier care poate fi încărcat în document cu această comandă. Acelaşi lucru îl putem face şi cu texte sursă. Mai mult, fişierele incluse cu această comandă pot include la rândul lor alte fişiere cu comanda `\input`.

Folosirea comenzii `\input` ajută la reducerea lungimii fişierelor cu care lucrăm (lucru important dacă dispunem de un editor care are o limită redusă a lungimii fişierelor cu care poate lucra).

Pentru a procesa doar o parte din documentul la care lucrăm putem folosi comezile `\include` şi `\includeonly`. Comanda `\include` este asemănătoare comenzii `\input`, în sensul că ea este practic înlocuită cu conţinutul fişierului inclus. Ea are un argument:

`\include{`*nume*`}`

unde *nume* este numele fără extensie al fişierului ce doriţi să-l includeţi. Acest fişier trebuie să aibă extensia `.tex`.

Atenţie! Dacă lucraţi sub sistemul de operare DOS, specificarea extensiei duce la distrugerea fişierului inclus!

Acest lucru se datorează faptului că pentru fiecare fişier inclus se scrie un fişier cu extensia `.aux`. De aceea, dacă specificăm ca *nume* fişierul `xxx.yyy`, `\include` caută fişierul `xxx.yyy.tex`, care în DOS este chiar `xxx.yyy` (prin trunchiere), şi îl citeşte. Va scrie fişierul `xxx.yyy.aux`, ceea ce în DOS duce la suprascrierea lui `xxx.yyy`, deci la pierderea iremediabilă a conţinutului acestuia.

Comanda `\include` poate apărea numai după `\begin{document}`, iar fişierele incluse cu ea nu pot conţine la rândul lor comenzi `\include`.

În general, se construieşte un fişier numit fişier rădăcină, care conţine preambulul şi cadrul document, în care apar doar comenzi `\include`.

Se poate folosi comanda

`\includeonly{`*nume1,nume2,...*`}`

pentru a specifica includerea doar a fişierelor *nume1,nume2,....* De exemplu, dacă în fişierul rădăcină (fişierul ce conţine comenzile `\include`) apar comenzile:

```
...
\includeonly{baze}
...
\begin{document}
...
\include{introd}
\include{baze}
...
```

atunci doar fişierul `baze.tex` va fi inclus, dar în aşa fel încât numerele de pagină, numerele ecuaţiilor, şi toate variabilele vor avea valorile pe care le-ar fi avut dacă ar fi fost incluse şi celelalte fişiere. Aceste informaţii sunt luate din fişierele `.aux` scrise pentru fiecare din fişierele incluse. Dacă aceste fişiere nu există sau nu mai sunt actuale, atunci aceste variabile nu vor avea valori corecte.

Comanda `\include` are un inconvenient: înainte şi după includerea fişierului se începe o pagină nouă. De aceea, este bine ca fiecare fişier astfel inclus să conţină unul sau mai multe capitole complete, deoarece oricum la începerea unui capitol se trece la pagină nouă.

Cu ajutorul acestor comenzi economisim mult timp, deoarece în general nu lucrăm la un moment dat decât la un capitol. Putem astfel procesa de oricâte ori numai acest capitol, fără să fie nevoie de reprocesarea celorlalte, la care oricum nu am modificat nimic. Înainte de a începe lucrul la alt capitol, procesăm întregul document, astfel încât fişierele `.aux` să fie actualizate.

Anexa B

Definirea opţiunii romanian

Opţiunile comenzii **\documentstyle** sunt de fapt pachete de definiţii TEX conţiunute în fişiere cu extensia **.sty**. De exemplu, opţiunea **makeidx** determină citirea fişierului **makeidx.sty**, care conţine în esenţă definiţia comenzii **\printindex**.

Putem defini alte opţiuni prin scrierea unui fişier cu extensia **.sty**. De exemplu la scrierea acestei cărţi am folosit un fişier numit **romanian.sty**, cu următorul conţinut:

```
\newcommand{\myqq@}{'}
\catcode'\'=13
\newcommand{'}[1]{\ifmmode {}^\prime \noexpand#1%
 \else\ifx#1a\u{a}%
 \else\ifx#1i\^{\i}%
 \else\ifx#1s\c{s}%
 \else\ifx#1t\c{t}%
 \else\ifx#1A\u{A}%
 \else\ifx#1I\^{I}%
 \else\ifx#1S\c{S}%
 \else\ifx#1T\c{T}%
 \else\ifx#1"\myqq@%
       \fi\fi\fi\fi\fi\fi\fi\fi\fi%
 \fi}
%
% Romanian Titles
%
\def\contentsname{Cuprins}
\def\listfigurename{Lista figurilor}
\def\listtablename{Lista tabelelor}
\def\figurename{Figura}
```

```
\def\tablename{Tabelul}
\def\partname{Partea}
\def\chaptername{Capitolul}
\def\refname{Referin'te}
\def\bibname{Bibliografie}
\def\appendixname{Anexa}
```

Prin acest fişier am modificat semnificaţia obişnuită a caracterului ' în aşa fel încât el să fie o comandă cu un parametru, folosit la generarea semnelor diacritice:

Iată un exemplu în care folosim noua definiţie a caracterului ' pentru a facilita scrierea textelor în limba română.

```
Iat'a un exemplu 'in care folosim noua
defini'tie a caracterului '" pentru a
facilita scrierea textelor 'in limba
rom\^an'a.
```

Acest lucru este ceva mai puţin obişnuit, şi nu poate fi făcut cu comenzi LaTeX. De aceea am fost nevoiţi să folosim comenzi TeX, pe care încercăm să le explicăm pe scurt aici.

În primul rând, ce înseamnă un caracter activ? Este un caracter căruia trebuie să-i dăm o definiţie, şi care va fi înlocuit peste tot unde apare cu definiţia lui. Deci practic este o comandă, şi poate avea sau nu parametri. În LaTeX există un caracter activ, şi anume caracterul ~. Acesta este o comandă fără parametru, care împiedică ruperea rândului în poziţia respectivă.

Fiecare caracter are un cod de categorie. Codul 13 îl au caracterele active, literele au codul 11, caracerele tipăribile care nu sunt litere au codul 12 etc.. Atunci când LaTeX citeşte fişierele de opţiuni, codul de categorie al caracterului @ este 11, adică el este considerat ca fiind literă. Acest lucru este util pentru a defini în aceste fişiere comenzi pentru uz local fără riscul ca acestea să fie redefinite în fişierul sursă LaTeX, la procesarea căruia codul de categorie al caracterului @ redevine 12.

Am folosit această facilitate pentru definirea comenzii `\myqq@` care să preia semnificaţia caracterului ', care îşi va pierde în continuare semnificaţia.

Prin linia `\catcode'\'=13`, caracterul ' primeşte codul de categorie 13. Urmează să dăm o definiţie comenzii astfel create. Conţinutul definiţiei se traduce în cuvinte astfel:

Dacă suntem în mod matematic, se pune un simbol „prim" urmat de parametrul comenzii care este nefolosit. Comanda `\noexpand` împiedică expandarea parametrului în caz că acesta este o comandă. Altfel, dacă parametrul este caracterul `a`, inserează comada `\u{a}`. Altfel, dacă parametrul este caracterul `i`, inserează comanda `\^\i` [Şi aşa mai departe]. Altfel, dacă parametrul este caracterul ", atunci inserează un apostrof prin comanda `\myqq@`. Comenzile `\fi` reprezintă sfârşitul multitudinilor de `\if`-uri.

Următoarele comenzi sunt redefinirea titlurilor folosite de LaTeX. S-a folosit comada TeX `\def`, deoarece unele secvenţe de control sunt definite doar la anumite stiluri, şi nu am şti unde să punem `\newcommand` şi unde `\renewcommand`. Comanda `\def` este mai brutală, ea nu face verificări asupra existenţei comenzii ce se defineşte.

Anexa C

Caractere ASCII

În această anexă prezentăm caracterele din fontul *roman* împreună cu codul ASCII (zecimal) asociat acestora, cât şi sursa LaTeX care le-a generat. Prin înlocuirea comenzii `\rm`, din sursă, cu o altă comandă de selectare a fontului (vezi tabelul **6.1** de la pagina 63) se pot obţine toate caracerele disponibile în LaTeX împreună cu codurile lor ASCII. Uneori este nevoie de aceste coduri pentru a putea fi transmise comenzii `\char`.

0 Γ	1 Δ	2 Θ	3 Λ	4 Ξ	5 Π	6 Σ	7 Υ
8 Φ	9 Ψ	10 Ω	11 ff	12 fi	13 fl	14 ffi	15 ffl
16 ı	17 ȷ	18 `	19 ´	20 ˇ	21 ˘	22 ¯	23 °
24 ¸	25 ß	26 æ	27 œ	28 ø	29 Æ	30 Œ	31 Ø
32 ␣	33 !	34 ”	35 #	36 $	37 %	38 &	39 ’
40 (	41)	42 *	43 +	44 ,	45 -	46 .	47 /
48 0	49 1	50 2	51 3	52 4	53 5	54 6	55 7
56 8	57 9	58 :	59 ;	60 ¡	61 =	62 ¿	63 ?
64 @	65 A	66 B	67 C	68 D	69 E	70 F	71 G
72 H	73 I	74 J	75 K	76 L	77 M	78 N	79 O
80 P	81 Q	82 R	83 S	84 T	85 U	86 V	87 W
88 X	89 Y	90 Z	91 [	92 “	93]	94 ^	95 ˙
96 ‘	97 a	98 b	99 c	100 d	101 e	102 f	103 g
104 h	105 i	106 j	107 k	108 l	109 m	110 n	111 o
112 p	113 q	114 r	115 s	116 t	117 u	118 v	119 w
120 x	121 y	122 z	123 –	124 —	125 ”	126 ˜	127 ¨

```
.....................
\newcounter{caractere}
.....................
{
\rm %Roman Style Characters
\setcounter{caractere}{0} % incepem cu caracterul cu codul 0
\whiledo{\value{caractere}<128}% pana la codul 127
{\noindent\makebox[1cm]{ %o cutie contine codul si caracterul:
\thecaractere\ \ \char \thecaractere}
\hspace{5mm}%5mm intre cutii.
\addtocounter{caractere}{1}%urmatorul caracter
}\
}
```

Anexa D

Stilul *letter*

Aşa cum am văzut în capitolul 2, LaTeX pune la dispoziţie şi un stil `letter` pentru scrierea de scrisori. Pentru aceasta în preambulul documentului se vor plasa următoarele comenzi:

- `\address{`*adresa*`}` comanda care poziţionează pe foaia de hârtie *adresa* expeditorului. Distribuţia pe rânduri a adresei se specifică cu comanda `\\` plasată acolo unde se doreşte trecerea la rând nou. Dacă această comandă lipseşte atunci scrisoarea va fi formatată pentru a fi copiată pe hârtie cu antet.

- `\name{`*nume*`}` sau `\signature{`*nume*`}` scrie la sfârşitul scrisorii, după formula de încheiere, numele *nume* al expeditorului. Dacă se doreşte extinderea pe mai multe rânduri atunci se va folosi ca argument şi comanda `\\`.

- `\telephone{`*telefon*`}` adaugă la adresa instituţiei, în josul paginii, parametrul *telefon* şi care de obicei reprezintă numărul de telefon al expeditorului. Această comandă are efect numai dacă din preambul lipseşte comanda `\address` (deci numai în cazul scrisorilor tipărite pe hârtie cu antet).

- `\location{`*loc*`}` adaugă la adresa standard a instituţiei, în josul paginii, parametrul *loc* care reprezintă de obicei numărul cabinetului etc. Această comandă are efect numai dacă din preambul lipseşte comanda `\address` (deci numai în cazul scrisorilor tipărite pe hârtie cu antet).

- `\date{`*data*`}` scrie sub *adresă* data specificată de parametrul *data* sau cea transmisă cu comanda `\today`. Comanda `\today` întoarce data

curentă aşa cum este ea setată în calculator (poate fi modificată în DOS cu comanda `date`).

- `\makelabels` are ca efect scrierea pe o foaie separată a etichetelor cu adresa destinatarului, foaie care mai apoi poate fi xerografiată pe etichete autoadezive. Comanda este opţională.

Scrisoarea ca atare este procesată în interiorul unui cadru special, cadrul `letter`, care se va afla obligatoriu în interiorul cadrului `document`. Sintaxa cadrului este următoarea:

`\begin{letter}{`*adresa_destinatar*`}`
`\opening{`*text*`}`
textul_scrisorii
`\closing{`*încheiere*`}`
`\ps{`*ps_text*`}`
`\cc{`*cc_text*`}`
`\encl{`*encl_text*`}`
`\end{letter}`

unde:

- *adresa_destinatar* reprezintă adresa destinatarului aşa cum va fi trecută pe etichetă de comanda `\makelabels`;

- *text* reprezintă formula de început a scrisorii şi va fi transmis ca parametru comenzii `\opening`;

- *textul_scrisorii* reprezintă de fapt conţinutul acelei scrisori şi este constituit din text ordinar împreună cu unele comenzi LaTeX (atenţie unele comezi îşi pierd semnificaţia— de exemplu `\chapter`, `\section` etc);

- *încheiere* reprezintă formula de încheiere a scrisorii şi va fi transmis ca parametru comenzii `\closing`;

- *ps_text* reprezintă textul care se transmite comenzii `\ps` şi care va fi plasat ca *Post Scriptum*. Dacă se doreşte ca pe foaie să apară şi iniţialele *P.S.* ele trebuie să facă parte integrantă din textul *ps_text*;

- *cc_text* reprezintă textul care se transmite comenzii `\cc` şi care este constituit în general din numele tuturor persoanelor cărora li s-au mai trimis copii ale acestei scrisori;

- *encl_text* este textul care se transmite comenzii `\encl` şi care reprezintă în general conţinutul plicului în care se află scrisoarea;

Comenzile `\ps`, `\cc` şi `\encl` sunt opţionale. Dacă nu se doreşte poziţionarea adresei destinatarului pe scrisoare, deci câmp *adresa_destinatar* vid, atunci, pentru a nu se obţine un mesaj de eroare cauzat de lipsa acestuia, se va înlocui *adresa_destinatarului* cu comanda `\mbox` cu argument vid (`\mbox{}`). Implicit paragrafele vor începe fără aliniat.

Anexa E

SLiTEX

SLiTEX este o versiune specială de LATEX destinată elaborării foliilor pentru retroproiector. Imprimarea textelor pe folie se poate face doar cu o imprimantă capabilă să tipărească pe un asemenea material.

Nu se recomandă folosirea LATEX-ului pentru scrierea foliilor, deoarece acestea necesită o formatare aparte. SLiTEX foloseşte fonturi diferite de LATEX. Fontul roman din SLiTEX este asemănător cu fontul sans serif din LATEX, dar dimensiunea normală în SLiTEX este aproximativ egală cu dimensiunea `\LARGE` din LATEX.

Un document ce urmează a fi procesat cu SLiTEX trebuie să fie compus din două fişiere: un fişier rădăcină şi un fişier cu textele pentru folii.

Pentru SLiTEX există un singur stil de document: `slides` , şi nu există opţiuni standard. Deci fişierul rădăcină va începe întotdeauna cu comanda `\documentstyle{slides}`. Preambulul va conţine numai declaraţii. Comenzile `\author`, `\title` şi `\date` nu au nici o semnificaţie.

Comanda `\colors` este destinată declarării culorilor care vor fi folosite. SLiTEX nu oferă facilităţi de tipărire pe imprimante color, dar poate prelucra texte color. Pentru fiecare culoare se tipăreşte o folie care conţine textele de culoarea respectivă. Prin suprapunerea acestor folii se va obţine rezultatul preconizat.

De exemplu, prin comanda

```
\colors{rosu,verde,albastru}
```

declarăm trei culori, numite `rosu`, `verde` şi `albastru`. SLiTEX nu cunoaşte semnificaţia acestor culori. Prin această declaraţie se definesc comenzile `\rosu`, `\verde` şi `\albastru`, care vor putea fi folosite în text la fel cum sunt

folosite comenzile de schimbare a fontului curent. Tot prin această declaraţie sunt definite cadrurile `rosu`, `verde` şi `albastru`. Dacă se vor produce doar folii alb-negru, nu este nevoie de această declaraţie.

După comanda `\begin{document}` se poate insera text care nu va fi însă formatat pentru folii, ci pentru a fi tipărit pe hârtie ca explicaţie despre foliile care urmează. Undeva între `\begin{document}` şi `\end{document}` trebuie să apară una din comenzile `\blackandwhite` sau `\colorslides`. Ambele au ca argument numele fără extensie al fişierului ce conţine textul foliilor. Acest fişier trebuie să aibă extensia `.tex`.

Fişierul al cărui nume apare ca argument în comenzile `\colorslides` sau `\blackandwhite` conţine unul sau mai multe cadruri `slide`:

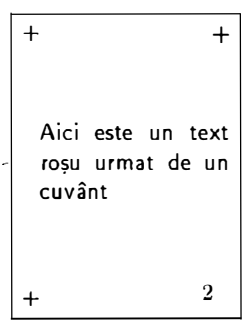
`\begin{slide}{`*lista de culori*`}`

...

`\end{slide}`

unde *lista de culori* este o listă de culori distincte declarate în `\colors`. Pentru fiecare din culorile din această listă se va genera o folie conţinând textul scris în culoarea respectivă. Dacă se produc folii alb–negru, lista trebuie să fie vidă.

Textul unui cadru `slide` trebuie să încapă pe o folie. SLITEX nu trece automat la altă folie.




```
\rosu
\begin{slide}{rosu,albastru}
Aici este un text ro\c{s}u urmat de
un cuv\^ant {\albastru albastru}.
\end{slide}
```

În exemplul de mai sus, imaginea din stânga prezintă folia roşie produsă de cadrul `slide` din dreapta.

Implicit, pe fiecare pagină sunt trasate trei cruciuliţe (utile la suprapunerea foliilor) şi numărul foliei în dreapta jos. Toate foliile corespunzătoare unui cadru `slide` vor avea acelaşi număr.

Stilul foliei poate fi modificat prin comanda `\pagestyle`. Există doar trei stiluri de pagină:

`headings` Cruciuliţele de aliniere şi numărul de folie sunt tipărite ca în exemplu. Acesta este stilul implicit.

`plain` Nu sunt tipărite cruciuliţele de aliniere, doar numărul de folie în dreapta jos.

`empty` Nu sunt tipărite nici cruciuliţe de aliniere, nici numărul foliei.

În preambul poate apare comanda `\onlyslides` care specifică ce folii vor fi generate. Comanda este utilă atunci când numărul total de folii este mare şi se fac corecturi doar la câteva. De exemplu:

```
\onlyslides{1,3-5,7-9999}
```

are ca efect generarea foliilor cu numerele 1, 3, 4, 5, 7 şi toate foliile care urmează (presupunând că numărul lor nu depăşeşte 9999).

În fişierul `.dvi` apar mai întâi toate foliile de prima culoare specificată în comanda `\colors` (în cazul nostru roşii), pe urmă toate cele din a doua culoare etc.. Fiecare set de folii este precedat de o pagină care specifică numele culorii care urmează.

Anexa F

REVTEX

Datorită răspândirii LaTeX-ului, a calităţii rezultatului şi nu în ultimul rând a faptului că sursa este în format text (pretându-se astfel la trimiterea ei prin poşta electronică), o serie de foruri ştiinţifice internaţionale au început să accepte articole trimise spre publicare în format LaTeX. Totuşi nu întotdeauna stilul propriu corespunde cu, să zicem, stilul `article` din LaTeX. Astfel au apărut alte noi stiluri specifice publicaţiilor respective. Cele mai cunoscute sunt $\mathcal{A}_{\mathcal{M}}\mathcal{S}$TeX al Societăţii Americane de Matematică şi REVTeX al Societăţii Americane de Fizică. Pe lângă adaptarea stilului la cel al revistelor de matematică, $\mathcal{A}_{\mathcal{M}}\mathcal{S}$TeX mai aduce şi o serie de noi fonturi şi simboluri matematice.

În prezent multe din universităţile din străinătate au definite diferite stiluri pentru crearea de lucrări de licenţă, lucrări de doctorat etc.

În general toate acestea sunt bazate pe LaTeX aducând în plus unele comenzi specifice sau modificând pe cele deja existente. Majoritatea acestor pachete fiind *shareware*, cum este cazul şi pachetului REVTeX la care ne vom referi în continuare. Un alt fapt important este acela că majoritatea pachetelor sunt însoţite de o amplă documentaţie.

REVTeX a luat fiinţă în urma eforturilor făcute de American Physical Society-APS (Societatea Americană de Fizică), Optical Society of America-OSA (Societatea de Optică a Americii) şi American Institute of Physics-AIP (Institutul American de Fizică). Reamintim că aceste societăţi publică unele dintre cele mai prestigioase reviste de fizică din întreaga lume.

Acest program se referă la toţi cei care scriu articole în *Physical Review A, B, C, D, E, Physical Review Letters, Journal of the Optical Society of America A, B, Applied Optics, Optics Letters, American Journal of Physics, Applied Physics Letters, Chaos, Computers in Physics, Journal of Applied*

Physics, The Journal of the Acoustical Society of America, The Journal of Chemical Physics, Journal of Mathematical Physics, Journal of Rheology, Journal of Vacuum Science and Technology A, B, Medical Physics, Publications of the Astronomical Society of the Pacific, Physics of Fluids A, B, Powder Diffraction şi *Review of Scientific Instruments.*

În continuare vom da doar acele comenzi care sunt în plus sau diferă faţă de stilul `article` din LaTeX (`#1` se referă la primul parametru ce va fi transmis unei comenzi, `#2` la al doilea etc.):

- `\documentstyle[#1]{revtex}` -specifică stilul documentului, iar `#1` reprezintă opţiunile. El poate lua valorile: `aps`, `pra`, `prb`, `prc`, `prd`, `pre`, `prl`, `preprint`, `tighten` (pentru APS şi AIP); `osa`, `manuscript` (pentru OSA şi AIP); `amsfonts`, `amssymb`, `eqsecnum`, `floats`, `10pt`, `12pt`. Pentru revistele publicate de AIP se acceptă, în general, oricare din opţiunile valide pentru APS şi OSA.

  - `aps` alege opţiunea pentru revistele publicate de APS;
  - `preprint` specifică formatul pentru preprint în cazul opţiunii `aps` (tipărire la două rânduri cu scopul de a trimite articolul, lăţimea fiind de 6,5 inch);
  - `tighten` opţiune pentru tipărirea la un rând atunci când este activă opţiunea `preprint`;
  - `pra,prb,...,pre,prl` opţiuni corespunzătoare revistelor APS la care se trimite articolul (`pra` Physical Review A, `prl` Physical Review Letters etc.);
  - `osa` opţiuni specifice revistelor publicate de OSA;
  - `manuscript` se foloseşte împreună cu `osa` şi este similar lui `preprint` de la opţiunea `aps`;
  - `amsfonts` opţiune care permite folosirea fonturilor Fraktur şi Blackboard Bold ale Societăţii Americane de Matematică (indiferent de celelalte opţiuni);
  - `amssymb` opţiune care împreună cu `amsfonts` permite folosirea simbolurilor matematice ale Societăţii Americane de Matematică (indiferent de celelalte opţiuni);
  - `eqsecnum` opţiunea are ca efect numerotarea ecuaţiilor pe secţiuni (indiferent de celelalte opţiuni);

  - `floats` în REVTEX tabelele şi figurile nu mai au caracter de flotanţi; astfel că pentru a permite acestora să se comporte ca în LATEX trebuie folosită opţiunea `floats` (nu se vor trimite articole cu această opţiune!);
  - `10pt`, `12pt` opţiuni de selectare a dimensiunii caracterelor normale la 10 respectiv 12 puncte tipografice (implicit este `11pt`);

- `\tighten` revine la tipărirea la un rând în cazul stilului `preprint` care (implicit) are ca rezultat tipărirea la două rânduri. Nu are efect când opţiunea `preprint` nu este activă;

- `\preprint{#1}` atunci când această comandă apare imediat după `\begin{document}`, are ca efect scrierea lui `#1` pe prima pagină în colţul din dreapta sus, dar numai cu opţiunea `preprint`;

- `\draft` omiterea acestei comenzii are ca efect netipărirea numărului de clasificare PACS (Physics and Astronomy Classification Scheme). Comanda se va plasa imediat după `\preprint`;

- `\title{#1}` aceeaşi semnificaţie ca şi în LATEX;

- `\author{#1}` aceeaşi semnificaţie ca şi în LATEX;

- `\address{#1}` aceeaşi semnificaţie ca şi în LATEX. Spre deosebire de LATEX comenzile `\author` şi `\address` vor fi scrise câte o pereche pentru fiecare autor;

- `\thanks{#1}` permite plasarea unor note de subsol referitoare la autor;

- `\pacs{#1}` scrie `#1` ca număr de clasificare PACS. Comanda va fi pusă întotdeauna (chiar dacă `#1` lipseşte) după `abstract`;

- `\narrowtext`, `\mediumtext`, `\widetext` tot textul care urmează va fi scris într-o coloană de lăţime 3,4 inch, 5,5 inch respectiv 7 inch;

- `\onlinecite{#1}` se va folosi, pentru a face citări în stil LATEX, numai la opţiunile în care comanda `\cite` scrie numărul bibliografic ca indice superior (de exemplu opţiunea `osa`);

- `\case{#1}{#2}` este echivalentă cu comanda LATEX `\frac`, doar că numărătorul `#1` şi numitorul `#2` vor fi scrişi în stilul `textstyle`. Opţional între `{#1}` şi `{#2}` se poate pune un `/`;

- `\lowercase{#1}` forţează scrierea lui `#1` cu litere mici. Comanda este necesară atunci când se doreşte scrierea unei litere mici în titlul unei secţiuni, deoarece titlul unei secţiuni este scris întotdeauna cu litere mari, ceea ce ar face imposibilă scrierea corectă în titlu a numelor elementelor chimice etc.;
- `\openone` produce caracterul $\mathbb{1}$;
- `\precsim`, `\succsim` produce caracterele $\precsim$ respectiv $\succsim$, în mod matematic;
- `\lesssim`, `\gtrsim` echivalete cu `\alt` respectiv `\agt`, produc simbolurile $\lesssim$ (*aproximativ mai mic*) şi $\gtrsim$ (*aproximativ mai mare*), în mod matematic;
- `\tensor{#1}` pune dublă săgeată deasupra lui `#1` (de exemplu comanda `$\tensor{x}$` produce $\overleftrightarrow{x}$);
- `\loarrow{#1}` pune săgeată la stânga deasupra lui `#1` (de exemplu `$\loarrow{x}$` produce $\overleftarrow{x}$);
- `\roarrow{#1}` pune săgeată la dreapta deasupra lui `#1` (de exemplu `$\roarrow{x}$` produce $\overrightarrow{x}$);
- `\overstar{#1}` pune steluţă deasupra lui `#1` (de exemplu comanda `$\overstar{x}$` produce $\overset{*}{x}$);
- `\overcirc{#1}` pune cerculeţ deasupra lui `#1` (de exemplu comanda `$\overcirc{x}$` produce $\overset{\circ}{x}$);
- `\Biggl`, `\biggl`, `\Bigl`, `\bigl` paranteze deschise în ordinea descrescătoare a mărimii lor;
- `\Biggr`, `\biggr`, `\Bigr`, `\bigr` paranteze închise în ordinea descrescătoare a mărimii lor;
- `\Bigglb`, `\bigglb`, `\Biglb`, `\biglb` paranteze îngroşate, deschise în ordinea descrescătoare a mărimii lor;
- `\Biggrb`, `\biggrb`, `\Bigrb`, `\bigrb` paranteze îngroşate, închise în ordinea descrescătoare a mărimii lor. Comenzile trebuie să apară în perechi (vezi şi comenzile LaTeX `\left` şi `\right`);

- `\corresponds` produce, în mod matematic, caracterul $\triangleq$;
- `\slantfrac{#1}{#2}` creează o fracţie înclinată în mod matematic. Nu va fi folosită pentru fişiere trimise la APS. De exemplu comanda `$\slantfrac{1}{2}$`, sau forma echivalentă `$\slantfrac{1}/{2}$`, produce fracţia $1/2$;
- `\lambdabar` generează, în mod matematic, simbolul lambda barat (λ);
- `\FL` aliniază la stânga ecuaţiile (nu funcţionează în stilul `preprint`);
- `\FR` aliniază la dreapta ecuaţiile (nu funcţionează în stilul `preprint`);
- `\eqnum{#1}` atribuie `#1` ca număr al ecuaţiei curente;
- cadrul `mathletters` permite *numerotarea* alfabetică a ecuaţiilor dintr-un grup de ecuaţii (vezi exemplul **6.1** de la pagina 128);
- cadrul `quasitable` defineşte un cadru `table` în text;
- `\acknowledgments` defineşte o secţiune pentru mulţumiri aduse persoanelor care au ajutat la crearea acelui articol;
- cadrul `references` defineşte un cadru ce este similar cadrului LaTeX `thebibliography`, cu care, de altfel, poate fi înlocuit;
- `\squeezetable` micşorează un tabel astfel ca el să încapă pe lăţimea paginii. Comanda se plasează imediat după comanda `\begin{table}`. Se va folosi în special cu opţiunea `osa` deoarece în cadrul ei tabelele au dimensiunea fixă de 15,25 cm;
- `\tableline` analogul comenzii `\hline` din LaTeX cu care, de altfel, poate fi înlocuită;
- cadrul `tabular` spre deosebire de LaTeX, cadrului `tabular` i s-a mai adăugat un format de coloană `d` care are ca efect alinierea numerelor fracţionare din acea coloană după punctul zecimal, celelalte fiind puse centrat;
- `\tablenote{#1}` este analoagă comenzii LaTeX `\footnote` şi permite plasarea unor note de subsol imediat după cadrul `table` (vezi exemplul **6.2** de la pagina 130);

- `\tablenotetext[#1]{#2}` este analoagă cu comanda `\footnotetext` din LaTeX şi permite specificarea textului `#2` al unei note de subsol la tabel al cărei număr este specificat prin parametrul opţional `#1`;
- `\tablenotemark[#1]` este analoagă comenzii LaTeX `\footnotemark` şi permite plasarea unui marker de notă de subsol acolo unde apare comanda;
- `\pounds` spre deosebire de LaTeX unde comanda are ca efect tipărirea semnului £, aici efectul ei este tipărirea caracterului $;
- `\bbox{#1}` permite scrierea lui `#1` cu stilul bold în mod matematic;
- `\Bbb` alege fontul *Blackboard Bold* în mod matematic;
- `\frak` alege fontul *Fraktur* în mod matematic;

1.EXEMPLE

$$X = x_1 + x_2 \qquad (1.1.a)$$

$$Y = y_3 + y_4 \qquad (1.1.b)$$

$$XY = X \cdot Y \qquad (1.1.c)$$

$$\begin{aligned} r &= z\log(\sqrt{x^2+y^2}) \qquad (1.1.d)\\ a &= b+c \end{aligned}$$

Aşa cum rezulta din ecuaţiile (1.1) ...
... ecuaţia (1.1.a), (1.1.b) ...

```
\section{EXEMPLE}

\begin{mathletters}\label{ec:cuplate}
\begin{equation}\label{ec:1}
  X=x_1+x_2
\end{equation}
\begin{equation}\label{ec:2}
  Y=y_3+y_4
\end{equation}
\begin{equation}\label{ec:3}
  XY=X\cdot Y
\end{equation}
\begin{eqnarray}\label{ecs:4}
r&=&z\log(\sqrt{x^{2}+y^{2}})\\
a&=&b+c          \nonumber
\end{eqnarray}
\end{mathletters}

A\c sa cum rezult\u a din
ecua\c tiile (\ref{ec:cuplate}) ...\\
... ec. (\ref{ec:1}), (\ref{ec:2})...
```

Figura 6.1: Exemplu de utilizare a cadrului `matheletters`.

În cadrul opţiunii `osa` există definite o serie de comenzi care permit scrierea foarte rapidă a multor nume de reviste de specialitate. Acestea sunt:

`\ao`	Appl. Opt.
`\ap`	Appl. Phys.
`\apl`	Appl. Phys. Lett.
`\apj`	Astrophys. J.
`\bell`	Bell Syst. Tech. J.
`\jqe`	IEEE J. Quatum Electron.
`\assp`	IEEE Trans. Acoust. Speech Signal Process.
`\aprop`	IEEE Trans. Antennas Propag.
`\mtt`	IEEE Trans. Microwave Theory Tech.
`\iovs`	Invest. Ophtalmol. Vis. Sci.
`\jcp`	J. Chem. Phys.
`\jmo`	J. Mod. Opt.
`\josa`	J. Opt. Soc. Am.
`\josaa`	J. Opt. Soc. Am. A
`\josab`	J. Opt. Soc. Am. B
`\jpp`	J. Phys. (Paris)
`\nat`	Nature (London)
`\oc`	Opt. Commun.
`\ol`	Opt. Lett.
`\pl`	Phys. Lett.
`\pra`	Phys. Rev. A
`\prb`	Phys. Rev. B
`\prl`	Phys. Rev. Lett.
`\pspie`	Proc. Soc. Photo-Opt. Instrum. Eng.
`\sjqe`	Sov. J. Quantum Electron.
`\vr`	Vision Res.

X	3.1456
x_d[a]	1235.4
x_2[b]	1456
x_3[b]	x_2

[a] prima notă.

[b] a doua notă.

```
\begin{table}
\begin{tabular}{|c|d|}
X&3.1456\\
\tableline
$x_{d}$\tablenote{prima not\u a}&1235.4\\
\hline
$x_{2}\tablenotemark[2]$&1456\\
\hline
$x_{3}$\tablenotemark[2]&$x_{2}$\\
\end{tabular}
\tablenotetext[2]{a doua not\u a}
\end{table}
```

Figura 6.2: Exemplu de utilizare a cadrului `table`.

Anexa G

Distribuţia emTEX

Specific acestei distribuţii[1] este posibilitatea de a desena linii şi modele care se repetă, folosindu-ne de comanda TEX `\special`. De asemenea putem include în text desene realizate cu alte utilitare (Gnuplot, PBrush[2], CorellDraw[3] etc). Fişierele care conţin astfel de desene trebuie să aibă obligatoriu formatul MSP sau PCX, dar aceasta nu mai este o problemă deoarece marea majoritate a utilitarelor de grafică au posibilitatea de a salva fişiere în aceste formate. Există chiar programe specializate care fac conversia dintr-un format în altul (de exemplu programul shareware *Graphic Workshop* (GWS) al firmei canadiene Alchemy Mindworks Inc.).

Comanda `\put` poate primi ca parametri următoarele comenzi TEX:

- `\special{em:graph` *fişier*`}` permite inserarea de desene realizate cu alte utilitare şi care se găsesc pe disc, în fişierul cu numele *fişier*, în format PCX sau MSP[4].
- `\special{em:message` *text*`}` textul *text* este afişat de către `dvidrv` şi apare după numărul de pagină înaintea parantezei de închidere `]`.
- `\special{em:point` *n*`}` defineşte un punct (*n=1÷32767*) şi obţine coordonatele punctului de referinţă.
- `\special{em:line` *n*`[h|v|p], ` *m*`[h|v|p][,`*lat*`]}` uneşte două puncte *n* şi *m*, definite cu comanda precedentă, cu o linie de lăţime *lat*.

[1] Lista fişierelor din distribuţie precum şi unele informaţii despre instalarea semiautomată a pachetelor se găsesc pe prima dischetă în fişierele `readme.*`, `help.*` şi `changes.*`.

[2] Marcă înregistrată a firmei Zsoft.

[3] Marcă înregistrată a firmei Corel Systems Corp.

[4] În acest caz fişierul `dvi` rezultant va depinde de rezoluţia folosită.

Punctele pot fi urmate de opţiunile `h,v,p` (care se exclud reciproc) care arată cum va fi tăiat capătul liniei (`h`-orizontal, `v`-vertical, p-perpendicular).

- `\special{em:linewidth` *lat*`}` setează lăţimea tuturor liniilor trase cu comanda precedentă la valoarea *lat*.

- `\special{em:moveto}` setează punctul curent fără a desena o linie. Coordonatele lui vor fi cele date ca argument comenzii `\put`.

- `\special{em:lineto}` desenează o linie între punctul curent şi cel de referinţă, punctul curent devenind apoi punct de referinţă. Coordonatele punctului curent sunt cele date ca argument comenzii `\put`.

În cazul pachetului LaTeX din distribuţia emTeX comanda `\put` mai poate primi ca parametru, în cazul în care s-a ales opţiunea `bezier`, şi comanda

- `\bezier{`n`}{`x_1, y_1`}{`x_2, y_2`}{`x_3, y_3`}` care construieşte curba *spline Bezier* în n puncte, definită de cele trei puncte (x_i, y_i i=1,2,3). Această comandă este mare consumatoare de memorie permiţând construirea de curbe Bezier de aproximativ 1000 de puncte.

G.1 `dvidrv`– utilitare pentru vizualizat şi tipărit fişiere `dvi`

Distribuţia de emTeX conţine în pachetul `dvidrv` şi următoarele utilitare[5]:

`dviscr` afişează pe ecran conţinutul unui fişier `dvi` (aşa cum ar apărea el pe hârtie) în cazul în care adaptorul grafic este CGA, EGA, VGA, HGC, iar procesorul este 80x86;

`dviscrs` afişează pe ecran conţinutul unui fişier `dvi` (aşa cum ar apărea el pe hârtie) în cazul în care adaptorul grafic este CGA, EGA, VGA, HGC, iar procesorul este 8086 sau 8088 (calculatoare mai lente cu memorie mică);

[5] Pe lângă utilitarele listate mai există: utilitar pentru creare de biblioteci de fonturi (`fontlib.exe`), utilitar pentru scris fişiere de configurare pentru imprimante matriciale `makedot.exe`, utilitar pentru convertit fişiere PCL (HP Laser Jet sau HP Desk Jet) în format MSP (Microsoft Windows Paint) sau PCX (`pcltomsp.exe`), o serie de fişiere de comenzi pentru diferitele tipuri de imprimante existente, precum şi fişiere cu documentaţie.

`dvivik` afişează pe ecran conţinutul unui fişier `dvi` (aşa cum ar apărea el pe hârtie) în cazul în care adaptorul grafic este Viking I;

`dvihplj` tipărire `dvi` la imprimantă HP Laserjet, HP Deskjet şi Kyocera F-1010;

`dvidot` tipărire `dvi` la imprimantă cu ace (configurabil cu ajutorul utilitarului `makedot`);

`dvimsp` utilitar pentru convertirea fişierelor `dvi` în fişiere grafice (format MSP sau PCX);

În cazul în care pachetul nu vine cu program de instalare (pachetul original nu conţine un astfel de program) este necesară modificarea valorilor implicite din fişierele `v.bat, vs.bat` în cazul în care acestea nu corespund cu tipul de fonturi ales (dependent de imprimantă). Implicit se presupune că se lucrează cu o imprimantă laser. Dacă imprimanta aleasă este —de exemplu— de tipul EPSON FX, atunci în fişierul `v.bat` va trebui să înlocuim `lj.cnf` cu `fx.cnf`. Fişierele de comenzi folosite pentru tipărirea la imprimantă a documentelor (`prt*.bat`) vor trebui modificate (sau create altele noi) numai în cazul în care imprimanta nu face parte din cele cunoscute de pachet (pentru cele cunoscute există deja fişiere de comenzi gata făcute — de exemplu pentru EPSON FX el se numeşte `prtfx.bat`).

Există următoarele tipuri de fişiere de configurare:

Nume	Rezoluţie	Nume Fonturi
fx.cnf	240x216	fx_*.fli
ito.cnf	160x144	ito_*.fli
p6l.cnf	180x180	p6l_*.fli
p6m.cnf	360x180	p6m_*.fli
p6h.cnf	360x360	p6h_*.fli
lj.cnf	300x300	lj_*.fli

În cazul în care imprimanta nu coincide cu cele cunoscute de pachet se va proceda astfel:

- se află din cartea tehnică a imprimantei rezoluţia maximă în mod grafic, iar apoi se alege un fişier `.cnf` corespunzător acelei rezoluţii (vezi tabelul de mai sus);
- cu ajutorul utilitarului `makedot` şi cu manualul imprimantei se scrie un fişier `.dot` specific acelei imprimante (vezi şi documentaţia aferentă lui `makedot`);

- se creează un nou fişier `.bat` având următoarea comandă
  `dvidot` *dotname* `@`*cnfname*`.cnf /po=prn %1 %2 %3 %4 %5 %6 %7`
  unde *dotname* este numele fişierului (deci fără extensie) `.dot` scris pentru imprimantă, iar *cnfname* este numele fişierului `.cnf` ales;

G.2 Modul de rulare al utilitarelor

- `numeutilitar ?` sau `numeutilitar ??` Afişează o parte din opţiuni (forma cu ?) sau toate opţiunile (forma cu ??). Aici `numeutilitar` reprezintă oricare din următoarele: dviscr, dvivik, dvimsp, dvihplj şi dvidot.

- `dviscr` [*opţiuni*] *fişier_intrare* [*opţiuni*]
  `dvivik` [*opţiuni*] *fişier_intrare* [*opţiuni*]
  `dvimsp` [*opţiuni*] *fişier_intrare* [*opţiuni*] *fişier_ieşire* [*opţiuni*]
  `dvihplj` [*opţiuni*] *fişier_intrare* [*opţiuni*] *fişier_ieşire* [*opţiuni*]
  `dvidot` *fiş_par* [*opţiuni*] *fişier_intrare* [*opţiuni*] *fişier_ieşire* [*opţiuni*]

Cu excepţia lui `dvidot` care trebuie rulat întotdeauna cu cel puţin un argument (*fiş_par*—numele fişierului de parametri `.dot`), celelate utilitare pot fi rulate fără argumente, cerându-le singure în timpul rulării.

Dacă parametrul *fişier_ieşire* lipseşte atunci ieşirea se face direct la imprimantă altfel în fişierul specificat. Ulterior acest fişier (binar !) poate fi trimis la imprimantă cu comanda:

`copy /B` *fişier_ieşire* `PRN`

Opţiunea `/B` este obligatorie deoarece în lipsa ei fişierul va fi tratat ca şi fişier text şi deci copierea se va opri la primul caracter `Ctrl Z` (end of text file— sfârşit de fişier text) întâlnit.

Oricare dintre opţiunile de mai sus pot fi înlocuite cu *@fişier_răspuns*; caz în care citirea opţiunilor se va face din acest fişier (*fişier_răspuns*). Lungimea liniilor din fişierul *fişier_răspuns* nu trebuie să depăşească 127 caractere.

De exemplu apelul utilitarului `dvidot` astfel:

```
dvidot fx100 @fx.cnf test.dvi
```

are ca efect citirea opţiunilor din fişierul răspuns *fx.cnf* şi tipărirea fişierului *test.dvi* la imprimantă.

G.3 Opţiuni comune tuturor acestor utilitare

În continuare caracterul:

reprezintă un număr care poate fi urmat de o unitate de măsură scrisă cu litere mici (de exemplu 10cm);

+ produce activarea unei opţiuni;

- produce dezactivarea unei opţiuni;

***** reprezintă orice alt tip de argument;

Toate opţiunile vor fi precedate de semnul - sau / şi vor fi separate, opţional, prin spaţii. În cazul alipirii opţiunilor, numai prima din şir poate să înceapă cu semnul - toate celelalte începând cu semnul /.

Între o opţiune şi argumentul său nu se admit spaţii; totuşi se admit semnele =, : şi :=. Astfel următoarele formulări sunt echivalente:
/t:1in
/t1in
/t=1in
/t:=1in
toate atribuind lui t valoarea de 1inch (25,4mm).

G.3.1 Opţiuni pentru afişarea de mesaje

/? afişează un mic mesaj de ajutor şi valorile implicite sau actuale ale opţiunilor necesare;

/?? afişează un mic mesaj de ajutor şi valorile implicite sau actuale ale tuturor opţiunilor;

/v# nivelul de eroare (implicit /v2)

/v0 numai mesajele de eroare fatală;

/v1 /v0 plus numerele de pagină şi substituirile de fonturi în mod *batch*;

/v2 /v1 plus avertismentele serioase;

/v3 /v2 plus toate avertismentele şi comenzile **\special**;

/v4 /v3 plus statistici şi numele fişierelor de fonturi;

/v5 /v4 plus comentariile din fişierele de fonturi;

G.3.2 Opţiuni de selectare a paginilor

/b* selectează prima foaie care va fi tipărită. Fişierul DVI va fi citit atunci de la început până se întâlneşte pagina potrivită. În total pot fi precizate 10 numere care vor fi comparate cu `\count0...\count9`. Numerele omise vor fi înlocuite cu un asterisc (ceea ce înseamnă orice număr de pagină). Opţiunea implicită este * care înseamnă începe cu prima pagină din fişier. De exemplu /b1.3.6 selectează paginile care au `\count0=1`, `\count1=3` şi `\count2=6`;

/e* selectează ultima foaie. După ce a găsit prima pagină potrivită, căutarea în fişierul DVI continuă până ce s-a găsit pagina care se potriveşte acestei opţiuni. Valoarea implicită este * —nu va avea loc nici o potrivire pentru ultima pagină. De exemplu /e12.3 înseamnă că ultima foaie este cea care conţine pagina cu numărul 12.3;

/2* tipărirea pe două feţe. /2o selectează numai foile impare (1.,3.,5.,...), /2e selectează numai foile pare (2.,4.,...). Opţiunea implicită este /2-, toate foile sunt selectate;

/k# numărul de foi (din domeniul ales) care for fi sărite la tipărire (implicit /k0 adică nu se sare nici o foaie);

/n# numărul maxim de pagini care va fi tipărit. Implicit oricât de multe;

/z+ tipăreşte paginile în ordine inversă începând cu ultima foaie selectată. Implicit /z-, tipăreşte prima pagină la început;

/np# tipăreşte # copii pentru fiecare foaie selectată (implicit 1). Fiecare foaie va fi copiată de # ori înainte de a trece la următoarea. Această opţiune nu este suportată de dviscr, dvivik şi dvimsp;

/nf# tipăreşte de # ori întregul fişier (sau numai domeniul selectat). Această opţiune nu este suportată de dviscr, dvivik şi dvimsp;

G.3.3 Opţiuni referitoare la font

/fb modul *batch* (comandă). Dacă un font nu este găsit atunci e căutat mai întâi în mărimea 1000, dacă totuşi nu este găsit atunci este căutat fontul cmr10 în mărimea cerută, dacă în urma acestei căutări fontul nu este găsit atunci se va căuta fontul cmr10 în mărimea 1000. Insuccesul ultimei căutări are ca efect generarea unei erori fatale şi ieşirea din program. Opţiunea poate fi anulată cu o opţiune /fc sau cu /fi;

/fc schimbă modul. Înainte de încărcarea unui font sau fişier grafic se permite schimbarea numelui acestuia. Opţiunea poate fi anulată cu o nouă opţiune /fc (deci este de tip comutator) sau cu /fi;

/fd# setează variabila internă *max_drift*. Dacă spaţierea liniilor sau/şi caracterelor nu este corectă se poate încerca setarea unei noi valori. Valorile uzuale sunt cuprinse între 0 şi 3. Valoarea implicită depinde de driver (se determină cu opţiunea ??);

/fi modul interactiv (implicit). Dacă un font sau un fişier grafic nu este găsit atunci se apelează la ajutorul utilizatorului. Opţiunea poate fi anulată cu o opţiune /fb sau cu /fi;

/fl# limita de încărcare a fonturilor. Specifică încărcarea tuturor caracterelor, dintr-un fişier de fonturi, care nu depăşesc # octeţi, odată cu încărcarea parametrilor fontului respectiv. O excepţie este opţiune /fl-1 care are ca efect încărcarea fonturilor la cerere (atunci când este nevoie de ele), testându-se, la început, doar existenţa fişierului cu fontul respectiv. Opţiunea implicită este /fl300. De exemplu: /fl0 – nu se încarcă, iniţial, nici un caracter;

/fs# /fsx# /fsy# scalarea fonturilor. Aceste opţiuni sunt folosite pentru reducerea dimensiunii unui font permiţând ca fonturile generate pentru echipamente de rezoluţie mai ridicată să fie folosite la rezoluţii mai scăzute. Reducerea rezoluţiei se realizează prin înlocuirea punctelor din dreptunghiul de lăţime /fsx# şi înălţime /fsy# cu un singur punct. Dacă dreptunghiul este pătrat atunci se va folosi opţiunea /fs# pentru a seta cele două dimensiuni la aceeaşi valoare. Valorile pot fi în domeniul 1 la 8. Rezoluţia dorită va fi setată cu opţiunea /rf#. Exemplu: /r180/fsx2/rfx360 înlocuieşte fonturile P6L (180x180) cu fonturi P6M (360x180) reduse.

G.3.4 Opţiuni referitoare la *layout*-ul paginii

/tr# transformări efectuate asupra paginii

/tr0 transformarea identică (dispunerea normală);

/tr1 rotaţie 90 de grade în sens trigonometric pozitiv;

/tr2 rotaţie 180 de grade;

/tr3 rotaţie 270 de grade în sens trigonometric pozitiv;

/tr4 reflexie faţă de diagonala \;

/tr5 reflexie verticală;

/tr6 reflexie faţă de diagonala /;

/tr7 reflexie orizontală;

Transformările 1...7 nu funcţionează pentru caractere foarte mari, sau cu comenzi `\special`. Transformările aplicate modelului $\begin{matrix} A & B \\ C & D \\ E & F \end{matrix}$ sunt

0	1	2	3
A B C D E F	B D F A C E	F E D C B A	E C A F D B

4	5	6	7
A C E B D F	E F C D A B	F D B E C A	B A D C F E

/h# fixează înălţimea paginii. Această valoare este folosită de `dviscr` şi de `dvivik` pentru a afişa marginile foii ca linii. `dvimsp` foloseşte această valoare pentru a fixa dimensiunea fişierului grafic (dacă opţiunea `/om+` nu este activă).

/l# setează marginea stângă la valoarea #. Implicit 1 inch.

/la# ajustează marginea stângă cu #. Mai exact dacă /la=x şi /l=y atunci pagina tipărită va fi mutată la dreapta cu y-x. Valoarea implicită depinde de driver.

/t# setează marginea de sus la valoarea #. Implicit 1 inch.

/ta# ajustează marginea de sus cu #. Mai exact dacă /ta=x şi /t=y atunci pagina tipărită va fi mutată în jos cu y-x. Valoarea implicită depinde de driver.

/w# fixează lăţimea paginii. Această valoare este folosită de `dviscr` şi de `dvivik` pentru a afişa marginile foii ca linii. `dvimsp` foloseşte această valoare pentru a fixa dimensiunea fişierului grafic (dacă opţiunea `/om+` nu este activă). Utilitarul `dvidot` reduce această valoare cu 1px dacă ea coincide cu valoarea maximă admisă.

G.3.5 Colajul

/c? afişează toate valorile opţiunilor /cx# şi /cy#.

/cf# valoarea de start pentru indicele de poziţionare (implicit 1). Indicele de poziţionare selectează perechile de opţiuni /cx# şi /cy# folosite la pagina curentă. Indicele este incrementat cu 1 pentru fiecare pagină; după ce ajunge la ultima valoare, indicele este repus pe 1. Valoarea finală este dată de /cp# sau este luat ca cel mai mare indice setat de /ci#. Pe prima pagină poziţiile 1 şi #-1 rămân nefolosite. De exemplu opţiunile /ci2/cx4in/cp2/cf2 duc la tipărirea a 2 pagini pe fiecare coală. Prima pagină pe jumătatea din dreapta a primei coli, a doua pagină pe jumătatea din stânga a celei de-a doua coli, a treia pagină pe jumătatea din dreapta a celei de-a doua coli, şi aşa mai departe.

/ci# selectează indicele pentru valorile /cx# şi /cy# următoare. Există o deplasare a paginii tipărite (care se datorează lui /l#, /la#, /t# şi /ta#) şi o deplasare a fiecărei pagini individuale care este tipărită pe aceeaşi coală. Se pot tipări până la 16 pagini pe o coală de hârtie, iar valorile deplasărilor pe x şi pe y (offset-ul) ale unei pagini componente sunt salvate în una din cele 16 poziţii selectate de # (valorile standard sunt în /cx0 şi /cy0). /ci fără nici un argument incrementează valoarea indicelui cu 1. Valoarea iniţială a indicelui este 1. Cea mai mare valoare a lui # —sau valoarea fixată prin /cp#, dacă este mai mare— determină numărul de poziţii de pagini diferite care există. De exemplu /ci2/cx4in duce la deplasarea la dreapta a paginii de pe poziţia a doua (a colii curente) cu 4 inch.

/cp# fixează numărul de pagini care vor fi tipărite pe o coală. Această valoare poate fi mai mică decât numărul poziţiilor de pagini, dar trebuie să fie un divizor al acestuia altfel rezultatul obţinut va fi ceva neobişnuit. Valoarea iniţială este 1. Dacă, de exemplu, există 4 poziţii pe pagină, cu opţiunile /cf1 /cp2, paginile vor fi dispuse pe coli astfel: pagina 1 pe coala 1 poziţia 1; pagina 2 pe coala 1 poziţia 2; pagina 3 pe coala 2 poziţia 3; pagina 4 pe coala 2 poziţia 4; pagina 5 pe coala 3 poziţia 1; şi aşa mai departe.

/cx#, /cy# deplasările (offset-ul) pe x şi pe y pentru pagina aflată pe poziţia selectată de /ci#.

De exemplu, opţiunile
/ci2/cx4in/ci3/cy4in/ci4/cx4in/cy4in/cp4

vor avea ca efect tipărirea pe o coală a 4 pagini (/cp4) în cele 4 poziţii specificate de cel mai mare argument al opţiunii /ci astfel: colţul din stânga sus (offset 0, 0) pagina 1, colţul din dreapta sus (offset 4in, 0) pagina 2, colţul din stânga jos (offset 0, 4in) pagina 3 şi colţul din dreapta jos (offset (4in, 4in) pagina 4 (vezi figura **7.1**).

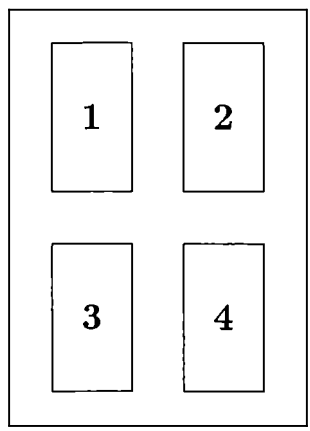


Figura 7.1: Tipărirea a 4 pagini pe o coală de hârtie.

G.3.6 Opţiuni referitoare la rezoluţie şi mărire

/m# dimensiunea din fişierul DVI este înlocuită cu #, care poate lua orice valoare între 100 şi 16000. Înainte de a folosi opţiunea trebuie să ne asigurăm că fonturile de dimensiunea aleasă sunt disponibile. De exemplu /m1200 înseamnă mărit de 1,2 ori (se împarte cu 1000), /m500 micşorat de 2 ori, /mh mărit de 1,095 ori iar , /m1414.213562 mărit de $\sqrt{2}$ ori.

/r#, /rx#, /ry# fixează rezoluţia, /rx# fixează rezoluţia orizontală, /ry# pe cea verticală, iar /r# pe amândouă odată. De exemplu /r180 semnifică rezoluţie verticală şi orizontală egală cu 180dpi (dots/inch). Folosirea unei anumite rezoluţii presupune existenţa pe disc a fonturilor corespunzătoare ca rezoluţie!

/rf#, /rfx#, /rfy# fixează rezoluţia fonturilor (în dpi). Această opţiune se va folosi, în general, împreună cu /fs#. /rfx# fixează rezoluţia ori-

zontală, /rfy# pe cea verticală, iar /rf# pe amândouă odată. Dacă opţiunea lipseşte se va folosi valoarea fixată pentru /r#.

/rg#, /rgx#, /rgy# fixează rezoluţia fişierelor grafice (în dpi). Această opţiune fixează valorile care vor fi folosite pentru $r şi $s în /pg*. /rgx# fixează rezoluţia orizontală, /rgy# pe cea verticală, iar /rg# pe amândouă odată. Dacă opţiunea lipseşte se va folosi valoarea fixată pentru /r#.

G.3.7 Opţiuni referitoare la directoare şi nume de fişier

/p@* directorul implicit pentru fişierul de răspuns. Dacă un astfel de fişier nu este găsit atunci el este căutat în lista dată ca parametru acestei opţiuni, dacă totuşi nici acum nu este găsit atunci este căutat în directoarele din variabila PATH.

/pd* directorul implicit pentru fişierul DVI. Dacă fişierul DVI nu este găsit atunci el este căutat în lista dată ca parametru acestei opţiuni. Valoarea implicită este `\emtex\texinput`.

/pf* directorul implicit pentru fonturi. Fonturile sunt căutate în lista dată ca parametru acestei opţiuni. În lista de directoare $s este înlocuit cu dimensiunea PXL (rezoluţia în dpi înmulţită cu 5), iar $r este înlocuit cu rezoluţia (dpi) — atenţie, ambele vor fi scrise cu litere mici. Directorul implicit depinde de driver (pentru imprimante laser el este `\emtex\pixel.lj\$rdpi`). Dacă se specifică mai multe căi ele trebuiesc separate prin ";". De exemplu opţiunea
`/pfc:\texfonts\pxl$s;c:\texfonts\dpi$r`
va căuta fontul `cmr10<300>` întâi în directorul `c:\texfonts\pxl1500` pe urmă, dacă nu l-a găsit, în directorul `c:\texfonts\dpi300`.

/pg* directorul implicit pentru fişiere grafice. Dacă fişierul grafic nu este găsit în directorul curent atunci el este căutat în lista dată ca parametru acestei opţiuni. Valoarea implicită este `\emtex\texinput`. Parametrii $s şi $r au aceeaşi semnificaţie ca la opţiunea /pf.

/pi* fişierul de iniţializare al imprimantei. Fişierul dat ca argument este trimis la imprimantă imediat după iniţializarea acesteia.

/pl* numele bibliotecilor de fonturi (create cu `fontlib`) care vor fi folosite. Fonturile vor fi căutate în biblioteci şi numai dacă nu sunt găsite sunt căutate ca fişiere separate.

/po* setează numele fişierului de ieşire. Dacă pe linia de comandă nu se specifică numele fişierului de ieşire atunci va fi folosit numele dat ca argument acestei opţiuni. Implicit se face ieşirea la imprimantă (deci nu există fişier pe disc pentru ieşire).

/ps* numele fişierului de substituire a fonturilor. Implicit este fără fişier de substituire a fonturilor.

/pt* numele fişierului log. Implicit se va folosi numele driverului plus extensia DLG (de exemplu, dvidot.dlg). Acest fişier va conţine toate mesajele afişate pe ecran plus valorile diferitelor opţiuni în timpul rulării respective. Fişierul poate fi folosit mai apoi ca fişier de răspuns pentru a se refolosi valorile fixate deja în timpul rulării anterioare. Opţiunea /pt (deci fără parametru) inhibă scrierea acestui fişier.

/pv* directoarele fişierelor VF (fonturi virtuale). Fonturile virtuale vor fi căutate numai în aceste directoare. Implicit nu se specifică nici un director (adică nu se caută fonturi virtuale).

/pw* scrie un fişier de răspuns pentru `fontlib`. El va fi un fişier text conţinând comenzile necesare lui `fontlib` pentru a insera fonturile folosite în timpul rulării într-o bibliotecă de fonturi. Pentru a genera numai fişierul de răspuns se va folosi şi opţiunea /n0. De exemplu /pw=new.cmd, după crearea fişierului new.cmd se va rula `fontlib` astfel: `fontlib new /k @new.cmd`.

G.3.8 Alte opţiuni

// aşteaptă introducerea de opţiuni (interactiv).

/u* fixează unitatea de măsură implicită care va fi folosită atunci când nu este specificată nici una. Pot fi folosite următoarele unităţi dimensionale (folosiţi numai litere mici):

pt punct tipografic (point=1/72,27in)

pc pica (12pt)

in inch (standard)

bp big point (1/72in)

cm centimetru

mm milimetru

dd didot point (1238/1157 pt)

cc cicero (12dd)

sp punct scalat (scaled point 1/65536pt)

px pixel (independent de rezoluţie)

G.4 Opţiuni suplimentare pentru `dviscr` şi `dvivik`

/- scris alb pe fundal negru.

/+ scris negru pe fundal alb (implicit).

/hx#, /hy# fixează poziţia iniţială (/hx0/hy0 reprezintă colţul din stânga sus).

/o#, /ox#, /oy# afişează cu scală de gri. Un dreptunghi de dimensiuni axb va fi afişat ca un singur punct (/o setează atât a cât şi b, iar /ox respectiv /oy setează doar a respectiv b). Este necesar cel puţin un display EGA cu 16 culori. Valorile lui a şi b pot fi întregi între 1 şi 8. Implementarea este optimizată pentru valorile 1,2,4 şi 8; pentru celelalte merge foarte încet.

/oa# precizarea adaptorului grafic. Valorile posibile sunt:

/oa0 autodetectare (implicit)

/oa1 CGA 640x200

/oa2 EGA 640x350 sau 640x200

/oa3 VGA 640x480

/oa4 HGC 720x348, numai sub DOS

/oa5 GENOA superEGA 800x600, numai sub DOS

/oa6 PARADISE VGA 800x600, numai sub DOS

/oa7 Video Seven VGA 800x600, numai sub DOS

/oa8 Tseng Labs EGA 800x600, numai sub DOS

/oa9 HP Vectra 640x400, numai sub DOS

/oa10 MDS Genius (PRIMUS II) 728x1008, numai sub DOS

/oa11 EGA 640x200, 16 culori

/oa12 EGA 640x350, 16 culori

/oa13 EGA 640x350, 16/64 culori

/oa14 VGA 640x480, 16 culori

/oa15 VGA 640x480, 16/64 culori

/oa16 VGA 640x480, 16/262144 culori

/oa17 EGA 640x350, 2 culori

/oa18 VGA 640x480, 2 culori

/oa19 Olivetti M24 640x400, numai sub DOS

Alte moduri particulare pot fi folosite cu următoarea extindere a acestei opţiuni: `/oa=m;w;h;ax;bx`; unde:

m este un număr care descrie combinaţia adaptor-ecran:

100 EGA monocrom

101 EGA 16 culori

102 EGA 64 culori

103 VGA cu ecran color analog

w este lăţimea ecranului în pixeli (de la 320 la 1024)

h este înălţimea ecranului în pixeli (de la 200 la 1024)

Modul grafic va fi setat prin încărcarea în registrul AX a valorii ax şi în registrul BX a valorii bx (dacă bx este 0 el poate fi omis) după care se apelează întreruperea 10H. Valorile ax şi bx vor fi date în hexazecimal. Dacă (w/8)*h depăşeşte 65536 atunci h va fi redus la (65536/w)*8. De exemplu /oa11 este echivalent cu /oa101;640;200;0010.

/oc# alege culorile care vor fi folosite (numai cu EGA sau VGA). Valorile posibile sunt de la 0 la 15, impicit este /oc2 (verde). Culorile pot fi alese separat astfel: /ocp# pentru marginea paginii, /ocr# rigla, ocs# linia de stare.

/or+ permite prezenţa reticulului gradat (implicit)

/os+ permite afişarea liniei de stare (implicit)

/ou* fixează unitatea de măsură folosită pentru gradaţiile reticulului şi pentru linia de stare (implicit pt).

/sx#, /sy#, /s# reduce dimensiunea paginii afişate astfel că pe ecran va încăpea o mai mare parte a unei pagini. Un dreptunghi de pixeli de /sx# pixeli lăţime şi /sy# pixeli înălţime va fi afişat ca un singur punct. Opţiunea /s# fixează atât x cât şi y la aceeaşi valoare. Valorile posibile sunt între 1 şi 8.

G.5 Opţiuni suplimentare pentru `dvimsp`

/- scris alb pe fundal negru.

/+ scris negru pe fundal alb (implicit).

/o* alege formatul fişierului grafic: există două formate MSP diferite— Windows 1.x (/o1) şi Windows 2.x (/o2, implicit). Fişierele PCX pot fi create cu /op.

/om+ creează cea mai mică figură posibilă. Marginile sunt făcute suficient de înguste astfel ca decupajul definit de /h# şi /w# să încapă fără risipire de spaţiu. Implicit este /om- adică fişierul va conţine o figură cu lăţimea dată de /w# şi înăţimea dată de /h#.

/ow+ lăţimea desenului este rotunjită pînă la următorul întreg divizibil cu 16px (implicit este /ow-).

G.6 Opţiuni suplimentare pentru `dvidot`

/ox+ foloseşte protoclul XON/XOFF pentru controlul transmiterii datelor pe porturile seriale (COM1 la COM4). Ieşirea se face direct la portul serial. În acest caz fişierul de ieşire va fi specificat ca nume de port (COM1, COM2, COM3 sau COM4). Dacă ieşirea nu se face pe porturile seriale sau avem /ox- atunci ieşirea se face folosind funcţii DOS.

Când se foloseşte fişier de parametri, dar imprimanta nu este aiw (Apple Imagewriter) sau itoh (C.ITOH 8510A):

/s+ două treceri pe fiecare linie. Dacă este selectat /s+ atunci fiecare linie este tipărită de două ori, prima dată tipăresc acele 1, 3, 5, ..., iar a doua oară tipăresc acele 2, 4, 6,.... Implicit avem /s-.

Când se foloseşte fişier de parametri pentru imprimantă aiw (Apple Imagewriter) sau itoh (C.ITOH 8510A):

/hf# fixează lungimea paginii. Dacă lungimea paginii este fixată la 0 (implicit) atunci se va lua ca lungime a paginii valoarea dată ca parametru opţiunii /h#.

/s+ modul lent de tipărire. Dacă este selectat /s+ atunci tipărirea se face unidirecţional (itoh.dot fixează de asemenea şi un mod de tipărire incremental). Implicit avem /s- care semnifică tipărire bidirecţională.

G.7 Opţiuni suplimentare pentru `dvihplj`

/oc# şterge toate fonturile temporare după # pagini. Se foloseşte atunci când nu există suficientă memorie pentru păstrarea tuturor fonturilor necesare. Implicit este 0 (nu şterge fonturile).

/od+ ieşirea se face la imprimantă HP DeskJet.

/of# fixează valoarea iniţială pentru numerele de identificare a fonturilor. Implicit este 0, valori posibile sunt de la 0 la 32736. Fonturilor încărcate în imprimantă li se vor atribui numere de la # la #+31.

/oi+ iniţializează imprimanta (implicit). Înaintea oricărei tipăriri imprimanta este iniţializată. Anularea opţiunii se face cu /oi-.

/oj+ deplasează paginile tipărite în sesiunea curentă, faţă de marginea tăvii de ieşire astfel ca paginile tipărite în două sesiuni distincte să poată fi separate uşor (deplasare diferită faţă de marginea tăvii).

/ok+ ieşirea se face pentru o imprimantă Kyocera F-1010.

/om# dimensiunea memoriei imprimantei în Kb (implicit 394). De exemplu /om1500 semnifică 1,5Mb de memorie disponibili.

/ox+ foloseşte protoclul XON/XOFF pentru controlul transmiterii datelor pe porturile seriale (COM1 la COM4). Ieşirea se face direct la portul serial. În acest caz fişierul de ieşire va fi specificat ca nume de port (COM1, COM2, COM3 sau COM4). Dacă ieşirea nu se face pe porturile seriale sau avem /ox- atunci ieşirea se face folosind funcţii DOS.

G.8 Taste funcţionale pentru `dviscr` şi `dvivik`

Q sau **Ctrl C** ieşire din program.

← mută fereastra cu jumătate din lăţimea ecranului la stânga (pentru paşi mai mici vezi F şi C).

→ mută fereastra cu jumătate din lăţimea ecranului la dreapta (pentru paşi mai mici vezi F şi C).

↑ mută fereastra cu jumătate din înălţimea ecranului în sus (pentru paşi mai mici vezi F şi C).

↓ mută fereastra cu jumătate din înălţimea ecranului în jos (pentru paşi mai mici vezi F şi C).

4 mută fereastra cu lăţimea ecranului la stânga.

6 mută fereastra cu lăţimea ecranului la dreapta.

8 mută fereastra cu înălţimea ecranului în sus.

2 sau spaţiu (spacebar) mută fereastra cu înălţimea ecranului în jos.

Home mută fereastra în poziţia iniţială (fixată cu opţiunile (/hx# /hy# sau cu H).

7 mută fereastra în colţul din stânga sus al paginii.

End mută fereastra în colţul din stânga jos al paginii (cel setat cu /hx# sau H).

1 mută fereastra în colţul din stânga jos al paginii.

Ctrl ← mută fereastra la marginea din stânga.

Ctrl → mută fereastra la marginea din dreapta.

9 sau **PageUp** pagina anteriară.

3 sau **PageDn** pagina următoare.

Ctrl PageUp prima pagină.

Ctrl PageDn ultima pagină.

I comutator imagine inversă.

D reîmprospătează imaginea ecranului.

H face poziţia curentă ca poziţie de început (home—vezi /hx#, /hy#, Home şi End).

F reduce pasul cu care se deplasează fereastra la apăsarea tastelor săgeţi (are 6 trepte).

C măreşte pasul cu care se deplasează fereastra la apăsarea tastelor săgeţi (are 6 trepte).

G merge la una din paginile selectate pentru a fi afişate şi o afişează (numărul de pagină este cerut în zona liniei de stare după apăsarea acestei taste).

O merge la pagina precedentă (ca afişare, nu ca număr de pagină) şi o afişează (driverul memorează ultimele 20 de pagini afişate). Va fi folosită după folosirea uneia din comenzile G, S, P, Ctrl PageUp sau Ctrl PageDn.

P afişează o pagină cu un număr de pagină dat (numărul de pagină este cerut în zona liniei de stare după apăsarea acestei taste). Căutarea se face începând cu pagina curentă spre sfârşitul fişierului. Numerele de pagină pot conţine carcterul * (de exemplu 1.*.2).

R comutator reticul.

Ctrl R comutator gradaţii pe reticul.

S caută şirul specificat (şirul de căutat este cerut în zona liniei de stare după apăsarea acestei taste, dacă acest şir conţine caractere invalide atunci se va emite un sunet). Căutarea se face de la începutul paginii curente până la sfârşitul fişierului, iar dacă şirul nu a fost găsit se caută dinspre început spre punctul curent. În momentul găsirii şirului se va poziţiona reticulul pe poziţia de început a acestuia.

W comută efectul tastelor săgeţi— este deplasată atât fereastra cât şi reticulul (dacă este prezent). Această comandă afectează cele 4 taste săgeată, Ctrl ←, Ctrl →, Home şi End. De asemenea este modificat (la a doua apăsare) efectul tastelor F şi C prin modificarea mărimii deplasării ferestrei.

U şi **Ctrl U** modifică unitatea de măsură a dimensiunilor. U efectuează o deplasare înspre înainte prin lista uzuală de 10 dimensiuni (vezi la /u* şi /ou*), iar Ctrl U efectuează o deplasare înspre înapoi.

+ şi − modifică scala de gri sau dimensiunea (zoom). Dacă este posibil se va modifica scala de gri; dacă nu, dimensiunea, dar cu păstrarea relaţiei dintre /ox, /oy sau /sx, /sy. Tasta + măreşte dimensiunea desenului, iar tasta − o micşoarează.

Alt G şi **Alt S** modifică scala de gri sau scara (zoom). Vor putea fi introduse unul sau două numere între 1 şi 8. Aceste numere vor fi considerate ca noi factori de scală de gri (Alt G) sau de scară (Alt S). Dacă se introduce un singur număr atunci el va fi luat ca valoarea lui x şi a lui y.

T dviscr/dvivik poate să schimbe transformările dinamic prin introducerea unui număr cuprins între 0 şi 7 (vezi /tr).

K comută coordonatele folosite pentru tastele săgeţi între coordonate ecran şi cele transformate.

G.9 Linia de stare pentru `dviscr` şi `dvivik`

Linia de stare (vezi figura **7.2**) afişează următoarele informaţii:

1. pasul selectat cu F sau C;
2. indică cine se deplasează la acţionarea tastelor săgeată (poate fi R, W sau nimic);
3. scala de gri (Gxy);
4. scara (Sxy);
5. numărul colii;
6. numărul paginii;
7. coordonatele reticulului gradat;
8. dimensiunea unei gradaţii pe reticul;
9. poate fi * sau nimic, şi arată dacă dvidrv lucrează sau este pregătit pentru a primi comenzi;
10. reticulul gradat;
11. marginile paginii;

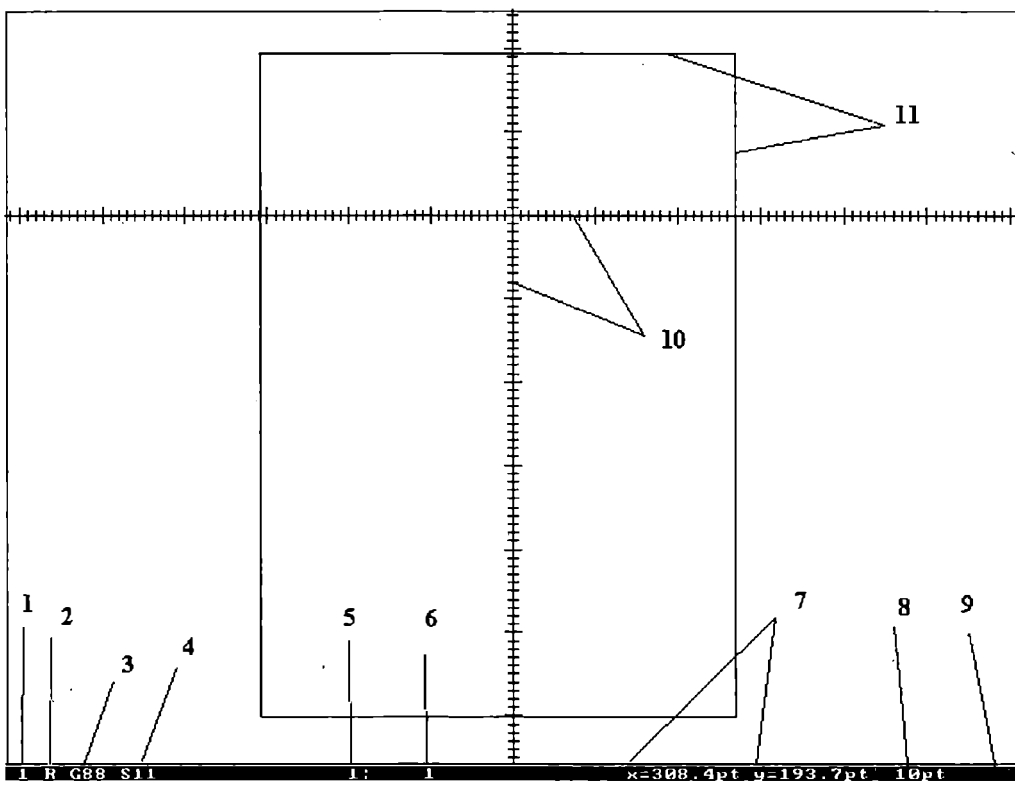


Figura 7.2: Linia de stare pentru `dviscr` şi `dvivik`

Index

\!, 82
\# (#), 63
#, 63
\$ ($), 64
$, 12, 64, 73
$$, 12, 38, 73
\% (%), 10, 64
%, 2, 64
\& (&), 64
&, 64
    în cadrul eqnarray, 39
    în cadrul tabular, 36
' (′), 74
' (pentru semne diacritice), 110
\(, 56, 73
\), 56, 73
\,, 68, 80
\-, 61
-, 68
--, 68
---, 68
\/, 68
\:, 80
\;, 82
< (¡), 64
> (¿), 64
\@, 68
@ in fisiere .sty, 110
\, 2, 10, 52, 64
\-, 16
\[, 73
\\, 66
    în \author, 20
    în \title, 19
\], 73
^, 12, 74
_ (_), 64
_, 64, 74
_, 12
\{, \} ({, }), 64
{, }, 64
    delimitatori de grup, 11
    delimitatori de parametri, 11
~, 66, 110
11pt, 18
12pt, 18

\␣, 10, 68

\abovedisplayshortskip, 97
\abovedisplayskip, 97
abstract, 22
abstract, cadrul, 28
\abstractname, 15
accente
    în mod matematic, 77
    în mod parafraf, 65
    obţinerea semnelor diacritice cu, 13
\acknowledgments (REVTEX), 127

\addcontentsline, 26
\address, 115
\address (REVTEX), 125
\addtocontents, 26
\addtocounter, 96, *114*
\addtolength, 89, 91
\addvspace, 69
\adjustarticle, 18
\adjustbook, 18
\adjustreport, 18
\Alph, 96
Alph, 94
\alph, 96
\alpha (α), 12
alph, 94
$\mathcal{A}_{\mathcal{M}}\mathcal{S}$, 123
$\mathcal{A}_{\mathcal{M}}\mathcal{S}$TEX, 123
\and
    în teste, 56
    în \author, 20
\appendix, 24
\appendixname, 15
\arabic, 96
arabic, 94
array, cadrul, 38, *39*, *76*
    lungimi pentru, 98
\arraycolsep, 98
\arrayrulewidth, 98
\arraystretch, 98
article, 18
\author, 20, *21*
\author (REVTEX), 125
.aux, 6

\backslash (\), 64
\baselinestretch, 91
\Bbb (REVTEX), 128
\bbox (REVTEX), 128
\begin, 27
\begin{document}, 19
\belowdisplayshortskip, 97
\belowdisplayskip, 97
\beta (β), 12
\bf, 11, 61
\bgroup, 54, *59*
\bibcite, 48
\bibitem, 6, 47
\bibname, 15
\Biggl (REVTEX), 126
\biggl (REVTEX), 126
\Bigglb (REVTEX), 126
\bigglb (REVTEX), 126
\Biggr (REVTEX), 126
\biggr (REVTEX), 126
\Biggrb (REVTEX), 126
\biggrb (REVTEX), 126
\Bigl (REVTEX), 126
\bigl (REVTEX), 126
\Biglb (REVTEX), 126
\biglb (REVTEX), 126
\Bigr (REVTEX), 126
\bigr (REVTEX), 126
\Bigrb (REVTEX), 126
\bigrb (REVTEX), 126
\bigskip, 69
\blackandwhite (SLITEX), 120
blank, *vezi* caracterul spaţiu
\bmod, 74, *75*
boldface, 61
\boldmath, 82
book, 17
\bottomfraction, 99
bottomnumber (contor), 99
bpt, 90

cadruri, 27–50
\cal, 62, 74, 82
caligraphic, 62

\caption, 44
    \ref folosit în, 56
caractere active, 110
caractere speciale, 9, 63
    comenzi de tipărire pentru unele, 9, 63
caracterul linie noua, 9
caracterul spaţiu, 9
\case (REVTEX), 125
\catcode (comandă TEX), 110
\cc, 116, 117
center, cadrul, *30*
\chapter, 23
chapter (contorul), 95
\chaptername, 15
\char, 113, *114*
\circle, 41
\circle*, 41
citate, 4, 28–29
\cite, 6, 47
\cleardoublepage, 68
\clearpage, 68
\closing, 116
cm, 90
cod de categorie, 110
\colors (SLITEX), 119
\colorslides (SLITEX), 120
comentarii, 2
comenzi, 2, 10
    numele unei, 52
    spaţii după, 52
comenzi fragile, 56
comenzi secţionale, 21–24
    forma cu *, 22
\contentsname, 14, 15
\corresponds (REVTEX), 127
cuprins
    scrierea unui, 14
    titlu pentru, 14
\dashbox, 41
\date, 20, *21*, 115
\dblfloatpagefraction, 99
\dblfloatsep, 100
\dbltextfloatsep, 100
\dbltopfraction, 99
dbltopnumber (contor), 99
\ddots (⋱), 74
\def (comandă TEX), 111
description, cadrul, *32*
    \item în, 32
despărţirea în silabe, 66
displaymath, cadrul, *37*
\displaystyle, 83
document, cadrul, 27
\documentstyle, 17, 28, 63
    opţiuni pentru, 18
    stiluri pentru, 17–18
\dotfill, 69, *70*
\dots (...), 74
\doublerulesep, 98
\draft (REVTEX), 125
.dvi, 7
dvi2ps, 8
dvidot, 7
dvihplj, 8
\egroup, 54, *59*
\em, 62
em, 90
em:graph, 131
em:line, 132
em:lineto, 132
em:linewidth, 132
em:message, 131
em:moveto, 132
em:point, 131
empty, 94
\encl, 117

\end, 27
\end{document}, 19
enumerate, cadrul, *32*
    \item în, 32
enumi (contorul), 95
enumii (contorul), 95
enumiii (contorul), 95
enumiv (contorul), 95
environment, *vezi* cadruri
eqnarray, cadrul, *39*
\eqnum (REVTEX), 127
\equal, 56
equation, *38*
    cadrul, 38
        lungimi pentru, 97
    contorul, 95
\evensidemargin, 91
ex, 90

\fbox, 85
\fboxrule, 86
\fboxsep, 86
\fi (comandă TEX), 110
figure, *45*
    cadrul, 44
        variabile pentru, 98
    contorul, 95
\figurename, 15
\fill, 90
\FL (REVTEX), 127
fleqn, 18
\floatpagefraction, 99
\floatsep, 99
\flushbottom, 67
flushleft, cadrul, *30*
flushright, cadrul, *30*
folie pentru retroproiector
    elaborarea unei, 119
font activ, 62
fonturi, 11, 61
\footnote, 71
    în minipage, 46
footnote (contorul), 95
\footnotemark, 21
\footnotesize, 63
\footskip, 91
\FR (REVTEX), 127
\frac, 12, *13*, 74
\frak (REVTEX), 128
\frame, 41
\framebox, *76*, 85
    în cadrul picture, 41
\frenchspacing, 68
\fussy, 66

german, 18
.glo, 7
\glossary, 50
\glossaryentry, 50
grup, 11, 52
\gtrsim (REVTEX), 126

\hat, *77*
\headheight, 91
headings, 94
\headsep, 91
\hfill, 69
\hrulefill, 69, *70*
\hspace, 11, 69, *70*
\hspace*, 69
\Huge, 63
\huge, 63
\hyphenation, 16

.idx, 7
\if (comandă TEX), 110
ifthen, 18, 56
\ifthenelse, *56*
\imath (ı), 77

minipage, cadrul, 46
\mit, 82
mm, 90
modul paragraf, 61
mpfootnote (contorul), 95
\multiput, 40
\myheadings, 94

\name, 115
\narrowtext (REVTEX), 125
\newcommand, 51, *52–54*, *59*
    efectul local al, 52
\newcounter, *59*, 96, *114*
\newenvironment, *57*, *59*
\newfont, 62
\newlabel, 25
\newlength, 89
newline, *vezi* caracterul linie nouă
\newline, 66
\newpage, 67
\newsavebox, 86, *87*
\newtheorem, *58*
\nocite, 47
\noexpand (comandă TEX), 110
\nofiles, 26
\nofrenchspacing, 69
\noindent, 69
\nolimits, 13
\nolinebreak, 66, 67
\nonumber, *39*
\nopagebreak, 67, 68
\normalmarginpar, 70
\normalsize, 63
\not, 56

\oddsidemargin, 91
\onecolumn, 95
\onlinecite (REVTEX), 125
\onlyslides (SLITEX), 121
\opening, 116
\openone (REVTEX), 126
operatori, 13
\or, 56
\oval, 41
\overbrace, 76, 77
\overcirc (REVTEX), 126
\overline, 76, *77*
\overstar (REVTEX), 126

\pacs (REVTEX), 125
page (contorul), 95
\pagebreak, 67, 68
\pagenumbering, 94
\pageref, 24, *25*
\pagestyle, 91
\par, 10, 69
parametri, 11, *13*
    comenzi cu, 11
\parbox, 46
\part, 23
part (contorul), 95
\partname, 15
pc, 90
picture, cadrul, 40, *43*
plain, 94
\pmod, 74, *75*
PostScript, 8
\pounds (REVTEX), 128
\precsim (REVTEX), 126
\preprint (REVTEX), 125
\printindex, 49, 109
\prod (∏), 13
\protect, *57*
prtfx, 8
prthplj, 8
\ps, 116, 117
ex, 90
\put, 40

in, 90
\include, 107, 108
\includeonly, 107, 108
\index, 7, 48, *49*
    parametrii pentru, 49
\indexentry, 7
\indexname, 15, 48
indici, *vezi* ^ şi _
\input, 55, 107
\int (∫), 13, 74
\intextsep, 99
\it, 61
italic, 61
itemize, cadrul, *31*
    \item în, 31

\jmath (ȷ), 77
\jot, 97

Knuth, Donald E., vii

\label, 6, 24, *25*
    eticheta ecuaţiilor cu, 38
\lambdabar (REVTEX), 127
Lamport, Leslie, 2, 5
\language, 16
\LARGE, 63
\Large, 63
\large, 63
\latex (LATEX), 10
LATEX, vii, 2
layout, 1
\ldots (...), 74
\left, 75, *76*
\left(, 13
\lefteqn, *39*
leqno, 18
\lesssim (REVTEX), 126
letter, 18
    cadrul, 116
    stilul, 115
\lim (lim), 74, *75*
\limits, 13, *39*, 77, *78*
\line, 41
\linebreak, 66
\linethickness, 40
list, cadrul, *33*
\listfigurename, 15
\listoffigures, 7, 25
\listoftables, 7, 25
\listtablename, 15
\loarrow (REVTEX), 126
\location, 115
.lof, 7
.log, 6
.lot, 7
\lowercase (REVTEX), 126

macrodefiniţii, 51–59
macrou, *vezi* macrodefiniţii
\makebox, 85, *114*
    în cadrul picture, 40
\makeglossary, 7, 50
makeidx, 18, 49, 109
\makeindex, 7, 48
makeindex (utilitar), 49
\makelabels, 116
\maketitle, 20, *21*
\marginpar, 70
\marginparsep, 91
\marginparwidth, 91
\markboth, 94
\markright, 94
\mathindent, 97
mathletters (REVTEX), cadrul, 127, *128*
\mbox, 65, *76*, 82, 85, *86*
\mediumtext (REVTEX), 125
\medskip, 69

\qquad, 82
\quad, 82
quazitable (REVTEX), cadrul, 127
quotation, cadrul, *29*
quote, cadrul, *5*, *28*

\raisebox, 85, 86
\ref, 6, 24, *25*
  folosit în \caption, 56
references (REVTEX)
  cadrul, 127
\refname, 15
\refstepcounter, *59*, 96
\renewcommand, *14*, 55, *56*, *59*
\renewenvironment, *57*
report, 18
reprocesare
  când este nevoie de, 8, 25, 47
\reversemarginpar, 70
REVTEX, 123
revtex (stilul), 124
\right, 75, *76*
\right), 13
\rm, 61
\roarrow (REVTEX), 126
\Roman, 96
Roman, 94
\roman, 96
roman, 94
roman, fontul, 61
romanian (opţiunea), 109
romanian.sty, 109
\ruggedbottom, 67
\rule, 71

\samepage, 68
sans serif, 61
\savebox, 86, *87*
\sbox, 86
\sc, 62
\scriptscriptstyle, 83
\scriptsize, 63
\scriptstyle, 83
\section, 23
section (contorul), 95
secvenţe de contol, *vezi* comenzi
semne diacritice, 110
\setcounter, 96, *114*
\setlength, 89
\settowidth, 89
\sf, 61
\shortstack, 41
showidx, 50
showidx, 18
\signature, 115
simboluri speciale, 65
\sl, 62
slanted, 62
\slantfrac (REVTEX), 127
slide, cadrul (SLITEX), 120
slides, stilul de document (SLITEX), 119
SLITEX, 119
\sloppy, 66
sloppypar, cadrul, 66
\small, 63
small caps, 62
\smallskip, 69
spaţiere, 80–82
\special (comandă TEX), 131
\sqrt, 74, *75*
\squeezetable (REVTEX), 127
\stackrel, 77
\stepcounter, 96
stiluri de document, 17–18
stiluri de numerotare, 94–95
stiluri de pagină, 91–94

\stretch, 89
\subsection, 23
subsection (contorul), 95
\subsubsection, 23
subsubsection (contorul), 95
\succsim (REVTEX), 126
\sum (∑), 13, 74, *75*
\symbol, 62

tabbing, cadrul, 34, *35*
    comenzi în, 34
\tabcolsep, 98
table, *45*
    cadrul, 43
      locul de amplasare pentru, 43
      variabile pentru, 98
    contorul, 95
\tableline (REVTEX), 127
\tablename, 15
\tablenote (REVTEX), 127, *130*
\tablenotemark (REVTEX), 128, *130*
\tablenotetext (REVTEX), 128, *130*
\tableofcontents, 6, 14, 25
tabular, cadrul, 35, *37*
    comenzi pentru, 36
    lungimi pentru, 98
    pentru REVTEX, 127
    specificatori de format pentru, 36
tehnoredactare, 4
tehnoredactor, 1
\telephone, 115
\tensor (REVTEX), 126
TEX, vii, 1, 2
\textfloatsep, 99
\textfraction, 99
\textheight, 91
\textstyle, 83
\textwidth, 91
\thanks, 21
\thanks (REVTEX), 125
thebibliography, cadrul, 46
\theequation, 97
theindex, cadrul, *48*
    \item în, 48
    \subitem în, 48
    \subsubitem în, 48
\thicklines, 40
\thinlines, 40
\thispagestyle, 94
\tighten (REVTEX), 125
\tilde, 77
\tiny, 63
\title, 19, *21*
\title (REVTEX), 125
titlepage, 18
.toc, 6
toc_entry, 18
\today, 10, 115
\topfraction, 99
\topmargin, 91
topnumber (contor), 98
\topsep, 97
totalnumber (contor), 99.
\tt, 11, 62
twocolum, 18
\twocolumn, 95
twocolumn, 70, 95
twoside, 18, 70
typewriter, 62

\unboldmath, 82
\underbrace, 76, *77*
\underline, 76
\usebox, 86, *87*

v, 7
\value, 56, 96
\vdots (⋮), 74
\vec, *77*
\vector, 41
\verb, *30*
\verb*, *30*
verbatim, cadrul, 30
verbatim*, cadrul, 30
verse, *29*
    cadrul, 29
\vfill, 69
\vspace, 69
\vspace*, 69

\whiledo, 56, *114*
\widehat, *77*
\widetext (REVTEX), 125
\widetilde, 77

xdvi, 7

Bibliografie

[1] **D. E. Knuth:** *The TEXbook,* Vol. A din seria *Computers and Typesetting,* Addison-Wesley Pub. Co. (1984), ISBN 0-201-13448-9.

[2] **L. Lamport:** LaTeX, *A Document Preparation System, User's Guide and Reference Manual,* Addison-Wesley Pub. Co. (1985), ISBN 0-201-15790-X.

[3] **H. Partl, Elisabeth Schlegl, Irene Hyna:** LaTeX-*Kurzbeschreibung,* EDV-Zentrum der TU Wien, Abt. Digitalrechenanlage (1990), H30.

[4] **H. Partl:** *Layout-Änderungen mit* LaTeX, EDV-Zentrum der TU Wien, Abt. Digitalrechenanlage (1988), H27.0.

[5] * * *: *The REVTEX Ver. 3.0 Input Guide*, Woodbury, New York, Washington D.C., 1992 (The joint Effort of APS, OSA & AIP).

[6] **L. Lamport:** *MakeIndex: An Index Processor For* LaTeX, documentaţie ce însoţeşte utilitarul `MakeIndex`.

[7] **Gabriela Luminiţa Bloju:** *Des-păr-ţi-rea în si-la-be,* Lucrare de licenţă, Univ. Timişoara, 1994.

Tipărit la
Editura şi Atelierele Tipografice ***METROPOL***

TEX este un sistem de culegere de texte destinat creării unor cărți frumoase – și în special a celor care conțin multă matematică.

– D. E. KNUTH, The TEXbook (1983)

De multe ori puterea nu vine atât din posesiune sau stăpânire ci din capacitatea de a face ceva ce alții nu știu sau nu pot. Astăzi ni se oferă posibilități nebănuite de a deveni mai puternici prin ceea ce facem cu ajutorul unui calculator bun și a unor programe inteligent concepute.

Această carte vă va familiariza cu cel mai performant program de formatat texte. Cu ajutorul lui veți putea face ca din imprimanta calculatorului dumneavoastră să apară texte editate conform celor mai exigente standarde de tehnoredactare.

Dacă nu veți avea pretenția unor tiraje mari, vă veți simți ca și cum ați stăpâni o editură și ați poseda o tipografie.

ISBN 973-31-0712-3

LEI 3800

A. PUSZTAI Gh. ARDELEAN • LaTeX Ghid de utilizare